



User Guide (version 2)

[Introduction to nuBuilder Forte](#)

[Navigating nuBuilder](#)

[Home Page](#)

[Globeadmin Home](#)

[User Home](#)

[Run Procedure](#)

[Run Report](#)

[Setup](#)

[Access Levels](#)

[User](#)

[Forms](#)

[Objects](#)

[Files](#)

[Fast Form](#)

[Procedures](#)

[Reports](#)

[Fast Report](#)

[SQL](#)

[Format](#)

[Translation](#)

[Database](#)

[Update](#)

[Browse Form](#)

[Edit Form](#)

[Builders](#)

[Fast Form](#)

[Fast Report](#)

[SQL Builder](#)

[Format Builder](#)

[Translation Builder](#)

[Customising nuBuilder Elements.](#)

[Forms](#)

[Objects](#)

[Arranging Objects](#)

[Reports](#)

[Arranging Objects](#)

[Procedures](#)

[Hash Cookies](#)

[Creating an Invoicing Application](#)

[Create Forms and Tables](#)

[Edit Form Properties](#)

[Arrange Objects](#)

[Edit Object Properties](#)

[Edit These Objects.](#)

[Create an Invoice Docket](#)

[Build SQL](#)

[Create Fast Report](#)

[Edit Report](#)

[Add Action Buttons](#)

[CSV Transfer](#)

[Exporting a CSV File](#)

[Importing a CSV File](#)

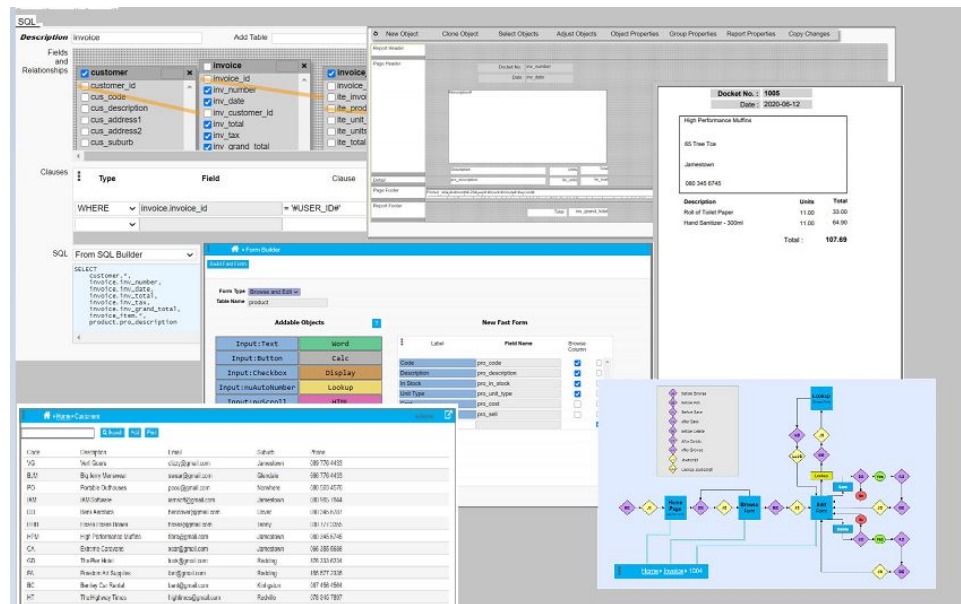
nuBuilder Forte is completely free and available for download from [Github](#) or [Sourceforge](#).

nuBuilder relies on support from its users.

If you would like to help support the ongoing development of nuBuilder you can do so here [Paypal](#).

Introduction to nuBuilder Forte

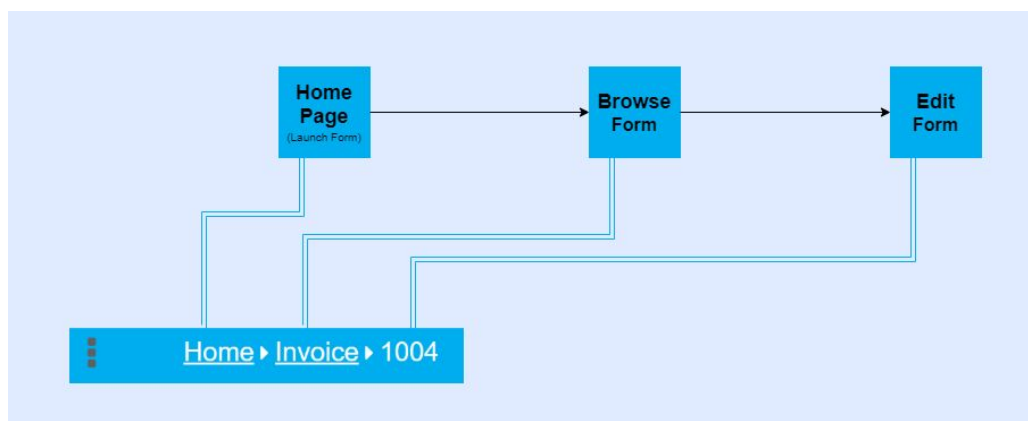
nuBuilder Forte is the 4th version of nuBuilder. It is web based, open source and is used for creating database applications with SQL, PHP and Javascript. All code used to customise is stored in the database and backed up



nuBuilder can be used to create business applications that manage data from a MySQL or MariaDB database. From **maintaining simple customer records** to **building a full accounting system**. This is done using Forms to search and edit data and Reports to print data.

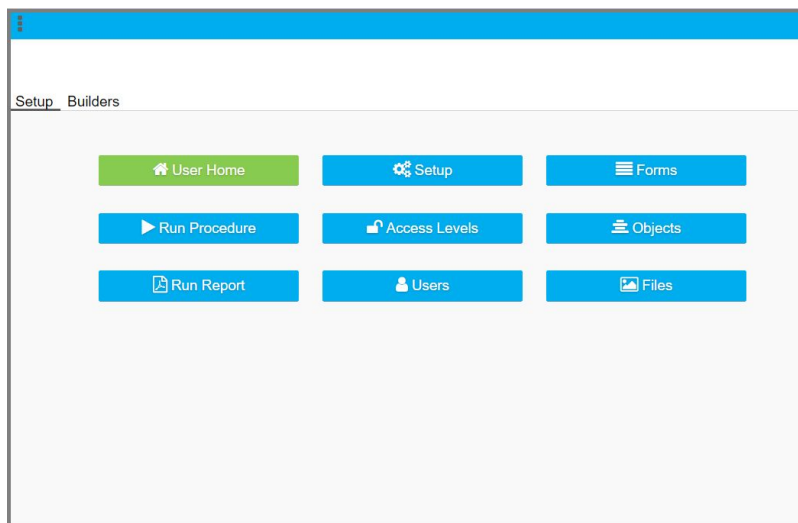
Navigating nuBuilder

nuBuilder Forte is easy to navigate. Each Form (or screen) is either a Browse Form or an Edit Form. When you first log into nuBuilder you will see a Home Page (Edit Form) appear in front of you. This Form will have buttons that can then take you to other Forms...



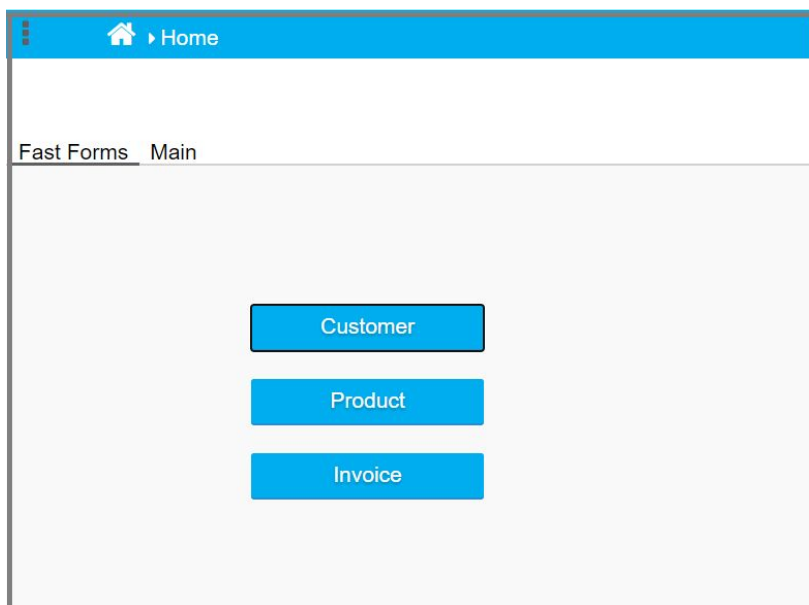
Home Page

Globeadmin Home



If a user logs in as globeadmin they will see a (Home Page) Edit Form with Tabs and Buttons that will lead to different elements of nuBuilder's development tools.

User Home



If a user logs in with any other username they will be taken to a Home Page created by globeadmin for users of other Access Levels. eg.

The User Home Button will take you to a Launch Form where Buttons are automatically placed when creating a **Browse / Edit** Form using Fast Form.

Run Procedure

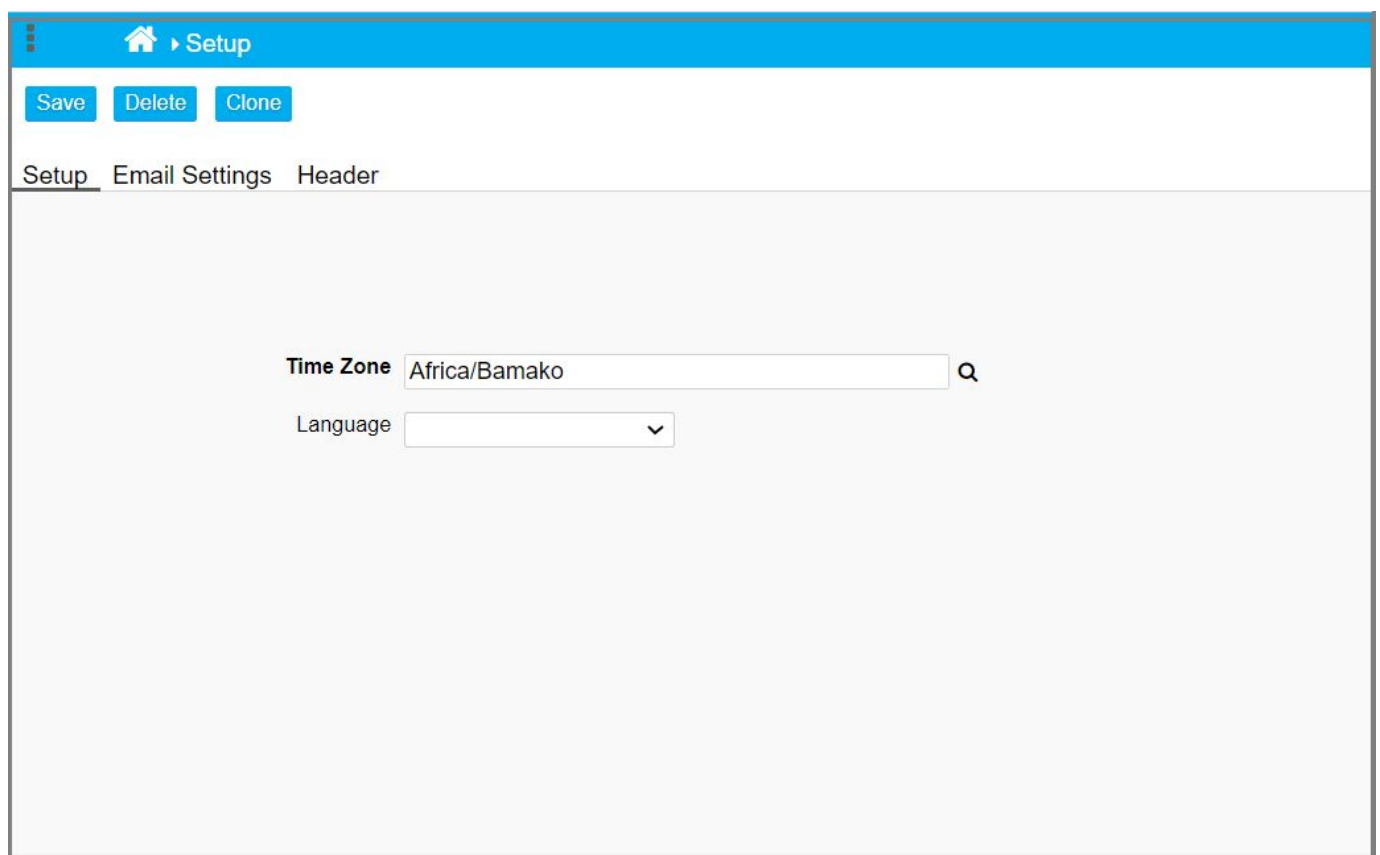
This Button gives Globeadmin access to the list of Procedures available to run.

Run Report

This Button gives Globeadmin access to the list of Reports available to run.

Setup

Click the Setup Button on the Setup Tab of the Home Form to edit the following...




The screenshot shows a web application interface for the 'Setup' section. At the top, there is a blue header bar with a home icon and the text 'Setup'. Below the header, there are three buttons: 'Save', 'Delete', and 'Clone'. Underneath these buttons, there are three tabs: 'Setup', 'Email Settings', and 'Header'. The 'Setup' tab is currently selected. The main content area of the 'Setup' tab contains two form fields. The first field is labeled 'Time Zone' and has a text input with the value 'Africa/Bamako' and a search icon (magnifying glass) to its right. The second field is labeled 'Language' and has a dropdown menu with a downward arrow icon.

Time Zone

Language

Set the language for the globeadmin user.

 Setup

Save

Delete

Clone

Setup Email Settings Header

Auth SMTP Information

User Name

bob@gmail.com

Password

1234

Host

smtp.gmail.com

Address

smtp.gmail.com

From Name

bob@gmail.com

Port

465

Use Authentication

Yes

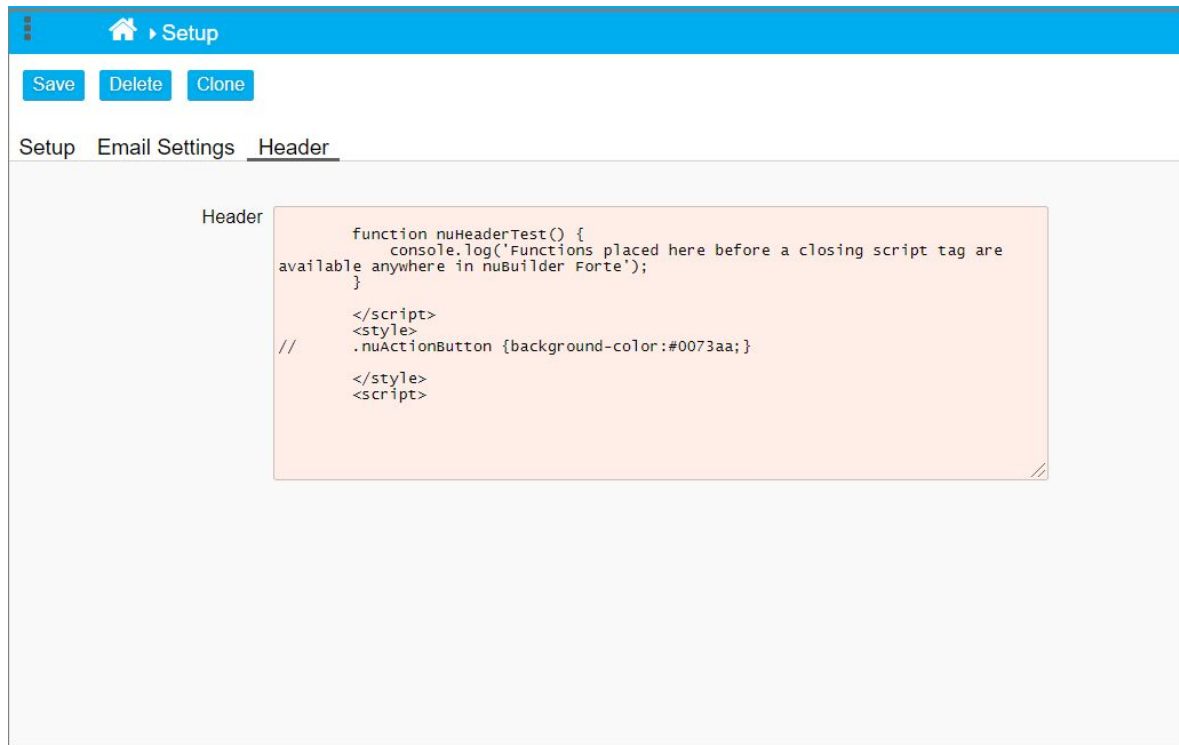
▼

Email Settings

Auth SMTP settings required...

- User Name
- Password
- Host
- From Address
- From Name
- Port
- Use SSL
- Use Authentication

Header



The Header is placed in the HTML5 header of nuBuilder Forte and can be used for adding...

- Javascript libraries
- Javascript functions
- Style Sheets
- CSS Styles

Eg.

```
<html>
  <header>
  <script>

    //- Any function in this script tag will be available anytime within nuBuilder.

    function nuHeaderTest(){

      console.log('Functions placed here before a closing script tag are available
anywhere in nuBuilder Forte');
    }

  </script>

  <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>

  <style>
    .nuActionButton { background : orange;          //-- personalise your Action Buttons
  </style>

  <script>
```


Access Levels

Once you have created your required Forms, Reports and Procedures you will need to give users access to them.

To do this you need to do 2 things

1. Create an Access Level.
2. Add a User.

Access Levels > A

Save Delete Clone

User Forms Procedures Reports

Home nuuserhome Q Home

Code A

Description Admin

Home

Choose the Form for this Access Level.

When you log in as a developer the first thing you see is the **Home Page**.

Each Access Level requires a **Home Page**, which will be its starting point for Users with this Access Level.

This **Home Page** should be a Launch Form.

The code for the default User Home Page is nuuserhome (the same as User Home on the Setup Tab).

Code

Lookup code for this Access Level

Description

Description of this Access Level

Forms

Home > Access Levels > A

Save Delete Clone

User Forms Procedures Reports

Form		Enable Buttons...					
		Add	Print	Save	Clone	Delete	
CUS	Customers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Add Forms that will be available to this Access Level.

Click the checkbox to enable the following on this Form for this Access Level.

- Add
- Print
- Save
- Clone
- Delete

Reports

The screenshot shows the 'Reports' tab selected in the 'Access Levels' configuration. The breadcrumb navigation at the top reads 'Access Levels > A'. Below the breadcrumb are three buttons: 'Save', 'Delete', and 'Clone'. The tab bar at the top of the main content area shows 'User', 'Forms', 'Procedures', and 'Reports', with 'Reports' being the active tab. The main content area contains a 'Report' dialog box. This dialog has a title bar with a menu icon and the word 'Report'. Inside, there is a search bar with the letter 'C' entered. Below the search bar, there is a list of reports. The first item is 'Customer List' with a search icon and an unchecked checkbox. The second item is partially visible with a search icon and a checked checkbox. A vertical scrollbar is on the right side of the list.

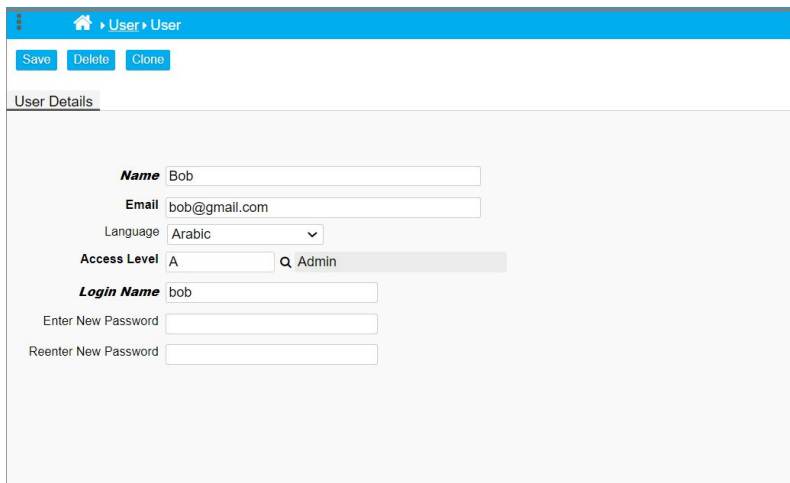
Add Reports that will be available to this Access Level.

Procedures

The screenshot shows the 'Procedures' tab selected in the 'Access Levels' configuration. The breadcrumb navigation at the top reads 'Access Levels > A'. Below the breadcrumb are three buttons: 'Save', 'Delete', and 'Clone'. The tab bar at the top of the main content area shows 'User', 'Forms', 'Procedures', and 'Reports', with 'Procedures' being the active tab. The main content area contains a 'Procedure' dialog box. This dialog has a title bar with a menu icon and the word 'Procedure'. Inside, there is a search bar that is empty. Below the search bar, there is a list of procedures. The first item is partially visible with a search icon and a checked checkbox. A vertical scrollbar is on the right side of the list.

Add Procedures that will be available to this Access Level.

User



The screenshot shows a web application interface for managing users. At the top, there is a blue header bar with a home icon and the text 'User > User'. Below the header, there are three buttons: 'Save', 'Delete', and 'Clone'. The main content area is titled 'User Details' and contains a form with the following fields:

- Name:** A text input field containing the value 'Bob'.
- Email:** A text input field containing the value 'bob@gmail.com'.
- Language:** A dropdown menu with 'Arabic' selected.
- Access Level:** A text input field containing 'A' and a search icon with the text 'Admin'.
- Login Name:** A text input field containing the value 'bob'.
- Enter New Password:** A text input field.
- Reenter New Password:** A text input field.

Name

User's real name

Email

Email address

Language

Select a language with a translation in Translations.

If left blank it will just use the phrases used in nuBuilder.

Access Level

Select a previously defined Access Level.

Login Name

nuBuilder Login Name

Enter New Password

nuBuilder Login Password

Only used when changing passwords.

Re Enter New Password

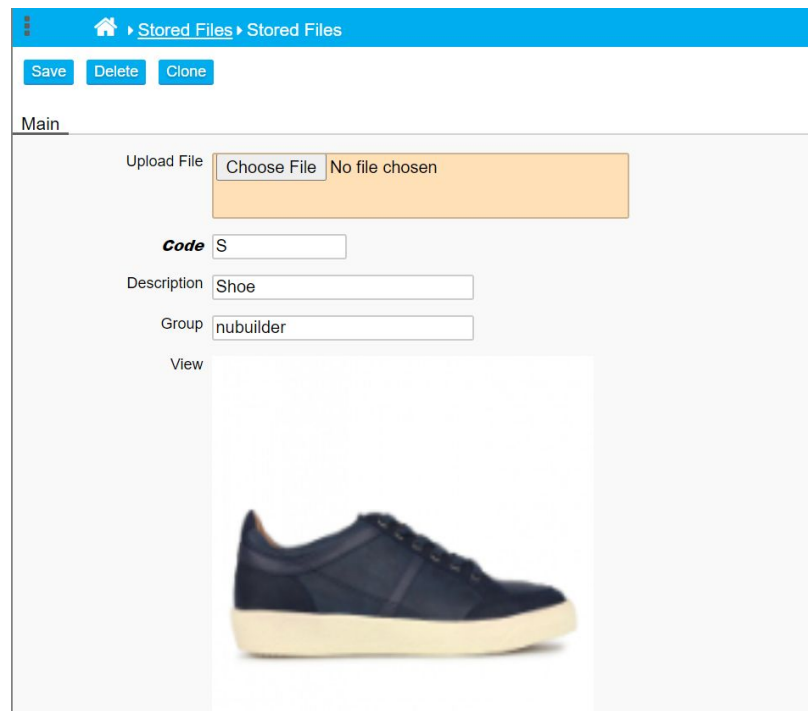
Repeat nuBuilder Login Password

Only used when changing passwords.

[Forms](#)

[Objects](#)

[Files](#)



nuBuilder Forte has an **Upload File** facility and an **Input:File** Object is to allow simple things like photos, pictures or logos. Because files are converted to JSON it's not practical to use this as a general method for storing bigger files over 300 Kb. As the conversion process can take a while.

Upload File

Upload or drag a file here.

Files are stored in nuBuilder Forte, in a JSON format that can be used in an HTML EMBED object...

```
{"file": "ZGF0YT...SlJVNUVya0pnZ2c9PQ==", "name": "nu_icon.png", "size": 21598, "type": "image/png"}
```

Code

Add a Lookup Code

Description

Add a Lookup Description

Group

Used to sort Images in a Browse Form.

Fast Form

Procedures

Reports

Fast Report

SQL

Format

Translation

Database

phpMyAdmin

Server: localhost | Database: information_schema

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central functions

Robert Favorites

New
information_schema
mysql
mysqltest

Tables
New

Table	Action	Rows	Type	Collation	Size	Overhead
customer	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	13	MySQL	utf_general_ci	2.4 K B	28 B
invoice_item	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	7	MySQL	utf_general_ci	2.4 K B	-
product	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	2	MySQL	utf_general_ci	2.4 K B	-
invoices	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	2	MySQL	utf_general_ci	2.2 K B	112 B
zzzzzys_access	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	4.8 K B	-
zzzzzys_access_form	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	4.3 K B	-
zzzzzys_access_report	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	9	MySQL	utf_general_ci	3.8 K B	-
zzzzzys_browse	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	4.3 K B	-
zzzzzys_cross_php	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	9	MySQL	utf_general_ci	15.5 K B	-
zzzzzys_debug	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	14	MySQL	utf_general_ci	16.7 K B	-
zzzzzys_event	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	10	MySQL	utf_general_ci	10.7 K B	-
zzzzzys_file	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	7	MySQL	utf_general_ci	14.9 K B	-
zzzzzys_form	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	40	MySQL	utf_general_ci	49.1 K B	-
zzzzzys_format	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	4	MySQL	utf_general_ci	2.3 K B	-
zzzzzys_object	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	10.2 K B	28 B
zzzzzys_php	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	13	MySQL	utf_general_ci	20.9 K B	-
zzzzzys_report	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	10.8 K B	-
zzzzzys_report_data	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Edit] [V View] [D Drop]	< 1	View	--	--	--
zzzzzys_run_list	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Edit] [V View] [D Drop]	< 1	View	--	--	--
zzzzzys_select_choose	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	2	MySQL	latin_swedish_ci	3.7 K B	-
zzzzzys_session	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	9	MySQL	latin_swedish_ci	2.8 K B	-
zzzzzys_setup	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	71.3 K B	46.5 K B
zzzzzys_tab	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	2.3 K B	-
zzzzzys_table	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	17	MySQL	utf_general_ci	8.9 K B	-
zzzzzys_timezone	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	28	MySQL	utf_general_ci	3.6 K B	-
zzzzzys_translate	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	170	MySQL	utf_general_ci	12.5 K B	-
zzzzzys_user	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1,208	MySQL	utf_general_ci	179.9 K B	-
zzzzzys_usr	[Star] [Browse] [JS Structure] [A Search] [I Insert] [E Empty] [D Drop]	1	MySQL	utf_general_ci	3.3 K B	-
Sum		~3,377	InnoDB	utf_general_ci	425.9 K B	44.5 K B

20 tables

☐ Check all / Check tables having overhead With selected

Print Data dictionary

Create table

Name: Number of columns: 4

phpMyAdmin will give nuBuilder developers direct access to the database.

It is a free software tool written in PHP, intended to handle the administration of MySQL over the Web.

phpMyAdmin supports a wide range of operations on MySQL and MariaDB.

Frequently used operations can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Those operations are..

- managing databases
- tables
- columns
- relations
- indexes
- users
- permissions

Update

Copied SYSTEM FILES to TEMP FILES
Copied SYSTEM FILES to TEMP FILES
Copied SYSTEM FILES to TEMP FILES for any new tables added from the import.
Updated TEMP FILE table structure's to SYSTEM FILES
Removed all ids starting with nu from TEMP FILES
Removed all ids not starting with nu from SYSTEM FILES
Inserted TEMP FILES into SYSTEM FILES
You will need to log in again for the changes to take effect.

Whenever a new update is pulled from [Github](#) it includes a file called nubuilder4.sql

This is used by the Update Button on the Builders tab on the Home Page, to make any changes to the current database.

The Update Button should be run each time files are pulled from Github or Sourceforge.

Browse Form

Clicking a button on the Home Page will generally take you to a Browse Form.

Breadcrumbs: Home > Invoice

Action Buttons: Search, Add, Print

Log out

Invoice No.	Date	Customer	Grand Total	Posted
1003	2020-06-02	Portable Outhouses	6715.50	No
1004	2020-06-02	Bens Aerobics	82.50	No
1005	2020-06-04	IAM Software	105.60	No
1006	2020-06-12	High Performance Muffins	107.69	No
1008	2020-06-09	Extreme Caravans	220.66	No
1008	2020-06-04	Hazelwood Pty Ltd	1326.49	No

Page Navigation: Page 1 / 1

- Breadcrumbs - Allows Users to retrace back through Previously opened Forms.
- Action Buttons - Actions that can be taken on this Form.
- Log out
- Search for strings - Filter records by searching for strings found in that record.
- Click to edit record - View this record on an Edit Form.
- Page Navigation - Move back and forth through Browse Form pages.
- Options.

Options

- ☐ Searchable Columns
- ☒ Form Properties
- ☒ Form Object List
- ☒ nuDebug Results
- ☒ Refresh
- ☒ Search
- ☒ Add
- ☒ Print

- Search Columns - Select only specific columns.
- Form Properties - View/edit this Form's properties.
- Form Object List - A Browse Form of all Objects on this Form.
- nuDebug Results - A Browse Form used for debugging.
- Refresh - Rebuilds this Form with the latest information from the server.
- Search - The same as hitting the Search Action Button.
- Add - The same as hitting the Add Action Button.
- Print - The same as hitting the Print Action Button.

Edit Form

Adding or selecting a record from a Browse Form will take you to an Edit Form.

Breadcrumbs: Home > Invoice > Invoice

Action Buttons: Save, Delete, Clone

Log out

Tabs: Main, Notes

Form Options: Date: 06-2020, Customer: MPM, Invoice No.: 1005

Subform Options: Items

Product	Unit \$	Units	Total	
HS Hand Sanitizer - 300ml	5.90	11.00	64.90	<input type="checkbox"/>
TP1 Roll of Toilet Paper	3.00	11.00	33.00	<input type="checkbox"/>
<input type="text"/>			0	<input checked="" type="checkbox"/>

Choose from another table's Browse Form

Subform Row Delete Checkbox

Total: 97.90
Tax: 9.79
Grand Total: 107.69

- Breadcrumbs - Allows Users to retrace back through Previously opened Forms.
- Action Buttons - Actions that can be taken on this Form.
- Log out
- Tabs.
- Choose from another table's Browse Form.
- Subform row delete checkbox.
- Form Options.

Options		
+	Arrange Objects	Ctrl+Shift+A
⚙	Form Properties	Ctrl+Shift+F
≡	Form Object List	Ctrl+Shift+O
🔍	nuDebug Results	Ctrl+Shift+D
↻	Refresh	Ctrl+Shift+R
💾	Save	Ctrl+Shift+S
🗑	Delete	Ctrl+Shift+Y
📄	Clone	Ctrl+Shift+C
?	Help	Ctrl+Shift+?

- Arrange Objects - Drag/ Drop and reorder Objects on the Edit Form.
- Form Properties - View/edit this Form's properties.
- Form Object List - A Browse Form of all Objects on this Form.
- nuDebug Results - A Browse Form used for debugging.
- Refresh - Rebuilds this Form with the latest information from the server.
- Save - The same as hitting the Save Action Button.
- Delete - The same as hitting the Delete Action Button.
- Clone - The same as hitting the Clone Action Button.
- Open any custom help.

- Subform Options.

Options	
+	Arrange Objects
⚙	Form Properties
≡	Form Object List
?	Help

- Arrange Objects - Drag/ Drop and reorder Objects on the Edit Form.
- Form Properties - View/edit this Form's properties.
- Form Object List - A Browse Form of all Objects on this Form.
- Open any custom help.

Builders

Fast Form

To create a Form that will allow for searching and editing records in a table, the easiest way is by creating a Fast Form.

A Fast Form can do the following...

- Create a **BrowseEdit** Form
- Use an existing table or create a new table in the database.
- Add fields to the Browse Form
- Add Objects to the Edit Form
- Create a button to access the new Form.

The screenshot shows the 'Form Builder' interface. At the top left is a 'Build Fast Form' button (7). Below it, the 'Form Type' is set to 'Browse and Edit' (1). The 'Table Name' is 'customer' (2). On the left, under 'Addable Objects', there is a list of input types and their corresponding objects (3). On the right, under 'New Fast Form', there is a table with columns for 'Label' (4), 'Field Name' (5), and 'Browse Column' (6). The table lists fields for the 'customer' table, with checkboxes for 'Browse' and 'Column'.

Label	Field Name	Browse	Column
Code	cus_code	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Description	cus_description	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Email	cus_email	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Phone	cus_phone	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Address 1	cus_address1	<input type="checkbox"/>	<input type="checkbox"/>
Address 2	cus_address2	<input type="checkbox"/>	<input type="checkbox"/>
Suburb	cus_suburb	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Postcode	cus_pc	<input type="checkbox"/>	<input type="checkbox"/>
		<input checked="" type="checkbox"/>	<input type="checkbox"/>

1. Select **Browse and Edit**.
2. Scroll through the list of available tables or add the name of the new table you want to create.

Repeat the following for adding each Object...

3. Click on an Addable Object.
4. Add the Object's Label
5. Scroll through the list of fields for the chosen table or add the name of the new field for a new table..
6. Check **Browse Column** if this **Field Name** is to appear in the Browse Form.

After adding all Objects...

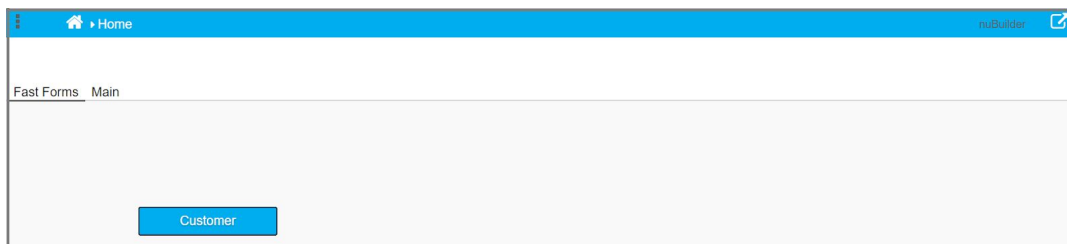
7. Click **Build Fast Form**.

This will result in the following message...



Creating this **Browse / Edit** Form with the Fast Form Builder will create the following...

A Button Object to open the new Form.



A Browse Form

Home Customers

Search Add Print

Code	Description	Email	Suburb	Phone
VG	Verti Goers	dizzy@gmail.com	Jamestown	089 776 4433
BJM	Big Jerry Menswear	swear@gmail.com	Glendale	666 776 4433
PO	Portable Outhouses	poou@gmail.com	Norwhere	080 563 4576
IAM	IAM Software	iamsoft@gmail.com	Jamestown	080 985 7844
DO	Bens Aerobics	bendover@gmail.com	Dover	090 345 6787
HHH	Hoses Hoses Hoses	hoses@gmail.com	Tenny	030 777 3355
HPM	High Performance Muffins	fibre@gmail.com	Jamestown	080 345 6745
CA	Extreme Caravans	xcar@gmail.com	Jamestown	066 355 5686
GD	The Pier Hotel	look@gmail.com	Redding	876 333 6234
FA	Freedom Art Supplies	fart@gmail.com	Redding	455 577 2335
BC	Bentley Car Rental	bent@gmail.com	Kinhgston	087 456 4564
HT	The Highway Times	hightimes@gmail.com	Redville	078 345 7897
HW	Hazelwood Pty Ltd	hazwood@gmail.com	Tonent	455 455 3567

Page 1 / 1

An Edit Form.

Home Customers Customers

Save Delete Clone

Main

Code IAM

Description IAM Software

Email iamsoft@gmail.com

Phone 080 985 7844

Address 1 20 Lake St

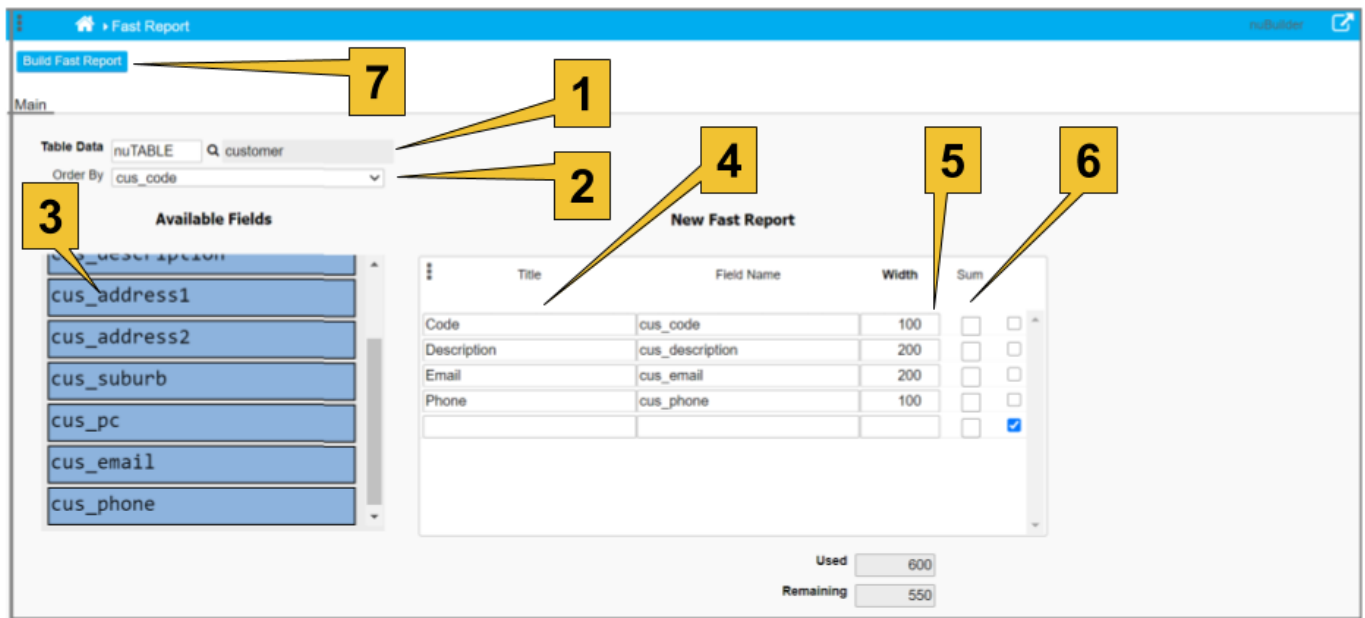
Address 2

Suburb Jamestown

Postcode 5123

Fast Report

To create a Report that can list and total fields, the easiest way is to use a Fast Report.



1. Select **Table Data**. (This is a list of table, SQL and Procedures that can be used)
2. Select the field to order the report by.

Repeat the following for adding each Object...

3. Choose **Addable Fields**.
4. Edit field title.
5. Adjust field width
6. Click to Sum this field on the report footer.

After adding all fields...

7. Click **Build Fast Report**.

This will result in the following message...



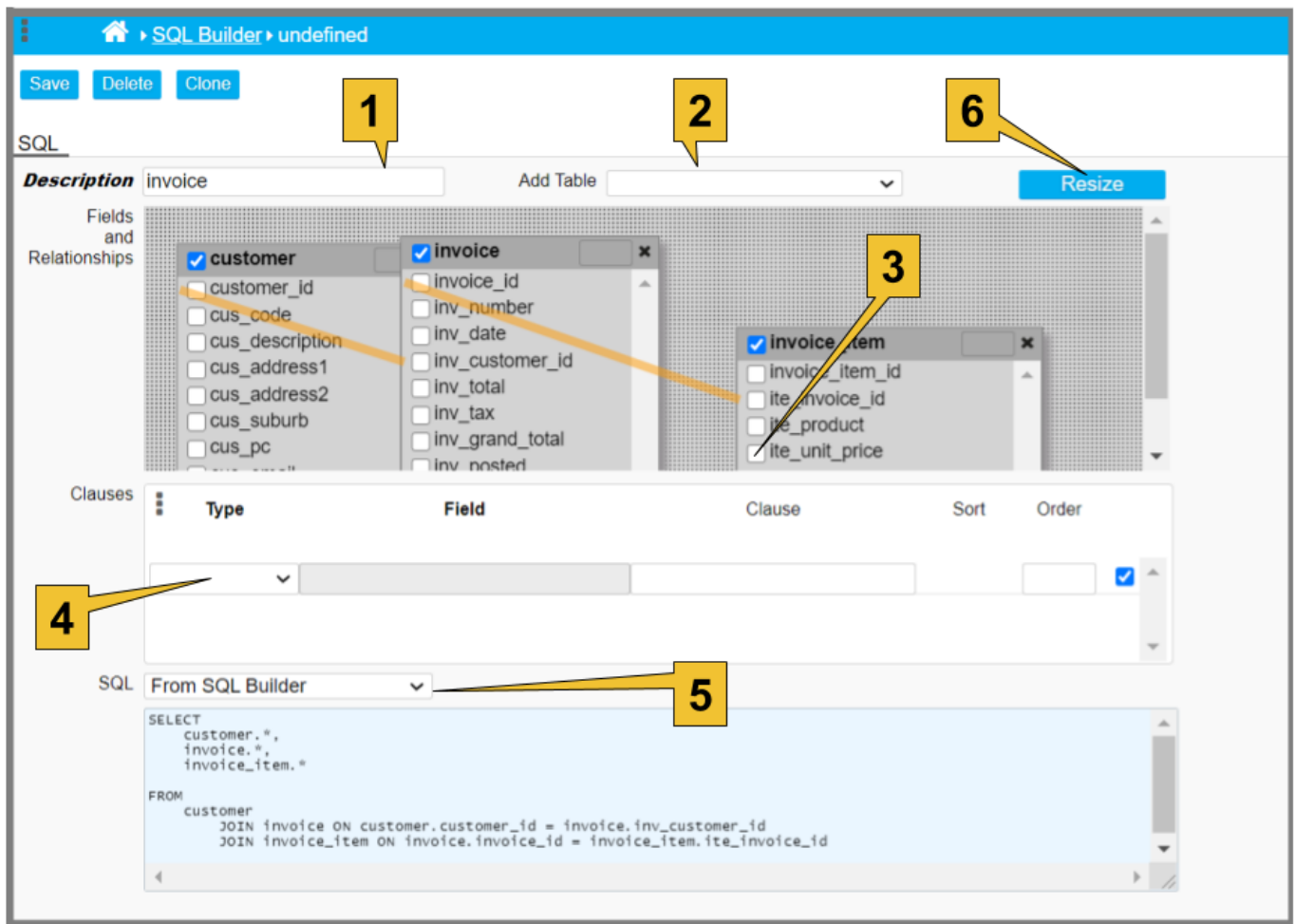
Creating this Report with the Fast Report Builder will create the following Report.

Fast Report 0			
Code	Description	Email	Phone
BC	Bentley Car Rental	bent@gmail.com	087 456 4564
BJM	Big Jerry Menswear	swear@gmail.com	666 776 4433
CA	Extreme Caravans	xcar@gmail.com	066 355 5686
DO	Bens Aerobics	bendover@gmail.com	090 345 6787
FA	Freedom Art Supplies	fart@gmail.com	455 577 2335
GD	The Pier Hotel	look@gmail.com	876 333 6234
HHH	Hoses Hoses Hoses	hoses@gmail.com	030 777 3355
HPM	High Performance Muffins	fibre@gmail.com	080 345 6745
HT	The Highway Times	hightimes@gmail.com	078 345 7897
HW	Hazelwood Pty Ltd	hazwood@gmail.com	455 455 3567
IAM	IAM Software	iamsoft@gmail.com	080 985 7844
PO	Portable Outhouses	poou@gmail.com	080 563 4576
VG	Verti Goers	dizzy@gmail.com	089 776 4433

Printed : 11-06-2020 09:57:59Page 1 of 1

Creating this Report with the Fast Report Builder will also create the following record in the **Report Builder** that can be further customised .

SQL Builder



The SQL builder only creates SQL SELECT statements.

These records can be used to create [Tables](#) used in [Reports](#).

The SQL Builder can also be accessed via the **SQL** button next to any nuBuilder Forte Textarea Object requiring SQL.

1. **Description** - Description for this SQL query.
2. **Add Table** - Choose tables to add to *Field and Relationships*.
3. **Field and Relationships** - This is a graphical way to create table relationships.

Each table can be dragged by clicking on the table name.

Each table added has...

- A Tablename.
- A field that will allow you to enter an alias for this table.
- A list of fields belonging to this table.
- A checkbox at the left of each field name allowing that field to be chosen.
- A checkbox at the top left allowing all fields to be chosen at once.

To create a relationship between 2 tables you need to do 2 things.

1. Click the first field as you hover over it and its name becomes red.
2. Unclick as you hover over a field on another table when its name becomes green.

This will create an orange line to represent this relationship.

When hovering over an orange line you will see its join type.

You can change this join type by double clicking it.

4. **Clauses**

Type

There are 4 types of clauses that can be added using the SQL Builder.

1. WHERE
2. GROUP BY
3. ORDER BY
4. HAVING

Field

This is a [scrolling](#) list of available field names.

Clause

Used by WHERE and HAVING. eg. = 1234

Hash_Variables can also be used. eg. = '#RECORD_ID#'

Order

Used by GROUP BY and ORDER BY. eg. **ASC or DESC**

5. **SQL** - SQL Code Built from *Graphical User Interface* that can be manually edited.

To further edit this, double click on the field and you will open [Ace Editor](#) in a full screen.

6. **Resize** - Resize *Fields and Relationships*.

Format Builder

Formats		
yy	yyyy	pp
dd	ddd	dddd
mm	mmm	mmm
hh	nn	ss
-	:	/
.	Space	,

The Format Builder will allow you to create custom formats for [nuDate](#) and [nuNumber](#).

To create a Format we need to do 2 things.

1. Choose the Data Type.
2. Build the Format.

Building a nuDate

To build a date format, click on a combination of the following options.

Entering a Date with the keyboard requires the user to enter the date in the exact format.

eg. *mask* - **result** (2007-13-01 15:20:01)

- yy - **17**
- yyyy - **2017**
- pp - **am** or **pm**
- dd - **21**
- ddd - **Tue**
- dddd - **Tuesday**
- mm - **03**
- mmm - **Mar**
- mmmm - **March**
- hh - **15**
- nn - **20**
- ss - **01**

.. and some punctuation.

Building a nuNumber

To build a nuNumber format you will need to provide up to 4 things.

1. A Currency sign (not essential)
2. A Separator that will separate a large number into thousands. (not essential)
3. A Decimal point character (only essential if Decimal Places is greater than 0)
4. Number of Decimal Places.

Changing these will change the display of the Format field.

All of these 4 fields are nuScroll Input Objects (with default values that can be scrolled through but will also accept other characters.)

Translation Builder

Translation > Translation

Save Delete Clone

Phrase

Language Arabic

English Adjust All Selected Objects Horizontally

Translation ضبط جميع الكائنات المحددة أفقياً

This allows you to make alternative phrases in other languages that can be used instead of nuBuilder Forte's English (Australian) phrases.

Language

The [User's](#) language.

English

A phrase used in nuBuilder Forte.

Translation

A phrase used in this language.

Customising nuBuilder Elements.

Forms

The first thing you will see after logging into nuBuilder Forte is a nuBuilder Launch Form containing Button Objects that will take you to other Forms.

This is also referred to as a Home Page.

There are 5 types of Forms.

1. Browse - A selectable list of records from a database table used by [Lookup Objects](#).
2. Edit - An area displaying editable fields from a record in a database table.
3. Browse and Edit - A Form to select and then edit a record in a database table.
4. Subform - An Edit Form that is repeated inside a Subform Object.
5. Launch - A Form not associated with a database table.

A Browse is used by a Lookup Object to select a record from another table.

You can build a Form (and create a [Button](#) to Access that Form) the easy way using the [Form Builder](#).

OR...

You can create a new Form the long way by...

1. Clicking the Form Button on the Setup Tab of the Home Page.
2. Clicking the Add Button at the top of a Browse Form.
3. Filling out your new Form's properties (see below). We suggest you make your first Form a [Browse and Edit](#).
4. Clicking the Save Button at the top of its Edit Form.
5. You will need to add a [Button](#) to the Home Page so users can open it.
6. And then you'll want to add some [Objects](#) to your new Edit Form, for displaying and editing fields.

nuBuilder Form › invoice

Save Delete Clone

Main Browse Custom Code

Form Type Browse and Edit

Code INV

Description Invoice

Table Name invoice

Primary Key invoice_id

Title	Order	Help
Main	10	<input type="checkbox"/>
		<input checked="" type="checkbox"/>

Form Type

Choose 1 of the following.

- Browse - Used by a Lookup Object.
- Edit - Used to display editable fields from a record in a database table.
- Browse and Edit - Used to find a record and then display it.
- Subform - Used inside a Subform Object.
- Launch Used...
 - As a Home Page
 - A Form to display other nuBuilder Objects
 - For adding criteria to a Report.

Code

A unique code of alphanumeric characters.

Description

Form description.

Table Name

The database table containing records and fields to display.

This is not required for Launch Forms.

Primary Key

The Primary Key belonging to the selected Table.

This is not required for Launch Forms.

Tabs

All Form types, apart from **Browse** Forms, require at least one Tab.

Tabs are arranged horizontally above the editable area of an Edit Form.

Tabs allow users to navigate (what seems like) different layers on the same Form.

Title

Tab Title

Order

Displayed order of the Tabs. - Every time the Form is saved these records this will be reordered and renumbered in multiples of 10.

Help

Javascript that can run to help the user eg. `window.open('www.help.com.')` or `nuMessage(['You must fill in','all fields.']);`

nuBuilder Form › invoice

Save Delete Clone

Main Browse Custom Code

SQL SQL Builder

SQL

```
SELECT
  invoice.*,
  customer.cus_description
FROM
  invoice
  JOIN customer ON customer.customer_id = invoice.inv_customer_id
```

Columns

Title	Display	Align	Format	Width	Order
Invoice No.	inv_number	Left		100	10
Date	inv_date	Left		100	20
Customer	cus_description	Left		150	30
Grand Total	inv_grand_total	Right	1000.00	100	40

Row Height Rows Per Page Redirect To

SQL

If you want any help building SQL, the SQL Button will direct you to nuBuilder's [SQL Builder](#).

This SQL is used to display records on a Browse Form

OR

used by Subform Objects to order records inside the Subform.

Columns

Title

Column Title

Display

Table Field Name

Align

Left, Right or Center

Format

Used for formatting dates and numbers - The Formatted part of results are not searchable. (Don't search for a \$ if it's not in the actual record.)

Width

Width of column

Order

Order of column. - Every time the Form is saved these records this will be reordered and renumbered in multiples of 10.

Row Height

Height in pixels

Rows Per Page

If '0' then all rows will be displayed, if a number fewer than the rows in the database, it will show the remaining rows on a separate page

Redirect To

Redirects the choice made this a Browse Form to another Edit Form.

nuBuilder Form › invoice

Save Delete Clone

Main Browse Custom Code

B Preview Browse Form **E** Preview Edit Form

PHP

BB Before Browse **BS** Before Save **BD** Before Delete

BE Before Edit **AS** After Save **AD** After Delete

Javascript

JS

PHP Code

PHP code can be added at the following places on a Form.

- **BB** - Before Browse
- **BE** - Before Edit - not available in Subform Forms.
- **BS** - Before Save - not available in Subform or Launch Forms.
- **AS** - After Save - not available in Subform or Launch Forms.
- **BD** - Before Delete - not available in Subform or Launch Forms.
- **AD** - After Delete - not available in Subform or Launch Forms.

These sections are explained in the flowchart below.

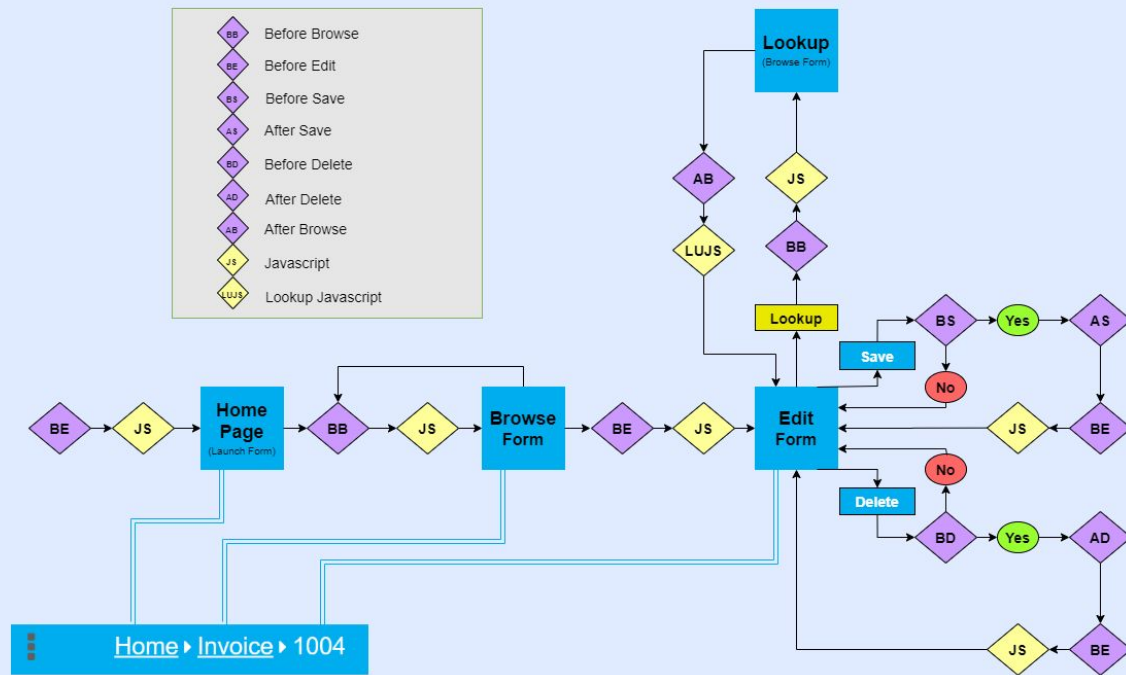
Javascript

A Textarea containing Javascript code - not available in Subform Forms.

To edit this, double click on the Textarea and you will open [Ace Editor](#) in a full screen.

This Javascript can be used on all Forms except for Subforms.

JS in the flowchart below.



Objects

Tab:All

Form Tab

Select a Form's Tab to add this Object to.

Type

Choose one of the following.

- Calc - A number field that recalculates after each change made on the current Edit Form.
- Display - A read only field that will display the first field of the first record from a SQL Select statement.
- HTML - A read only area that can be used to display html.
- Image - A read only area that will display an image.
- Input - Generally used for entering text, it can also be used for other HTML5 Input types, along with some types exclusive to nuBuilder Forte.
- Lookup - A lookup allowing the selection of a value from another table.
- Run - Does 1 of 2 things.
 1. Creates a Button to allow the user to run another Form, Report or Procedure.
 2. Creates an iFrame on an Edit Form in which a Form, Report or Procedure is displayed.
- Select - Allows the selection of one or more values from a list.
- Subform - Can contain rows (Subform Forms) that relate to the Edit Form.
- Textarea - Allows for text with carriage returns.
- Word - A read only object that simply creates a phrase in Bold.

Label

Description of Object.

ID

A valid HTML element id that is unique to this Edit Form.

Top

Value in pixels.

Left

Value in pixels.

Width

Value in pixels.

Height

Value in pixels.

Cloneable

Yes or No.

If set to No on an invoice number Object.

It would mean that the whole record is cloned but its invoice number is not.

Used by...

- Calc
- Input
- Lookup
- Select
- Textarea

Align

Left, Right or Center.

Used by...

- Calc
- Display
- Input
- Textarea
- Word

Validation

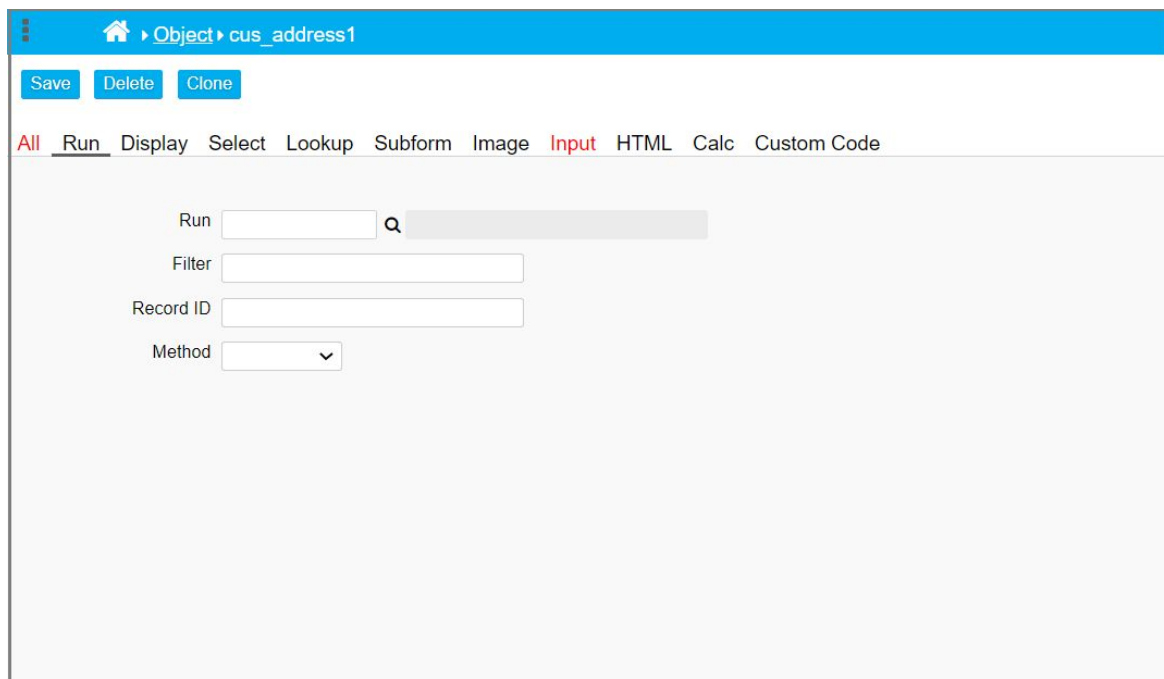
None, No Blanks or No Duplicates.

Used by...

- Input
- Lookup
- Select
- Textarea

Access

Editable, Readonly or Hidden.



The screenshot shows the Object Editor interface for an object named 'cus_address1'. The top bar is blue with a home icon and the breadcrumb 'Object > cus_address1'. Below the bar are three buttons: 'Save', 'Delete', and 'Clone'. A horizontal menu contains tabs: 'All', 'Run' (selected), 'Display', 'Select', 'Lookup', 'Subform', 'Image', 'Input' (highlighted in red), 'HTML', 'Calc', and 'Custom Code'. The main area contains four input fields: 'Run' with a search icon, 'Filter', 'Record ID', and 'Method' with a dropdown arrow.

Tab:Run

This Object will either...

- Create a Button that will open a Form, Report or Procedure in a new [Breadcrumb](#).
- Display a Form, Report or Procedure, inside an iFrame on the Edit Form.

Run

Choose Form, Report or Procedure to run.

Filter

A string that will be used like the **Search** field to filter Browse records.

[Hash Cookies](#) can be used in this string.

Record ID

Entering a Record ID will make the record with the matching ID display in an Edit Form (skipping the Browse Form)

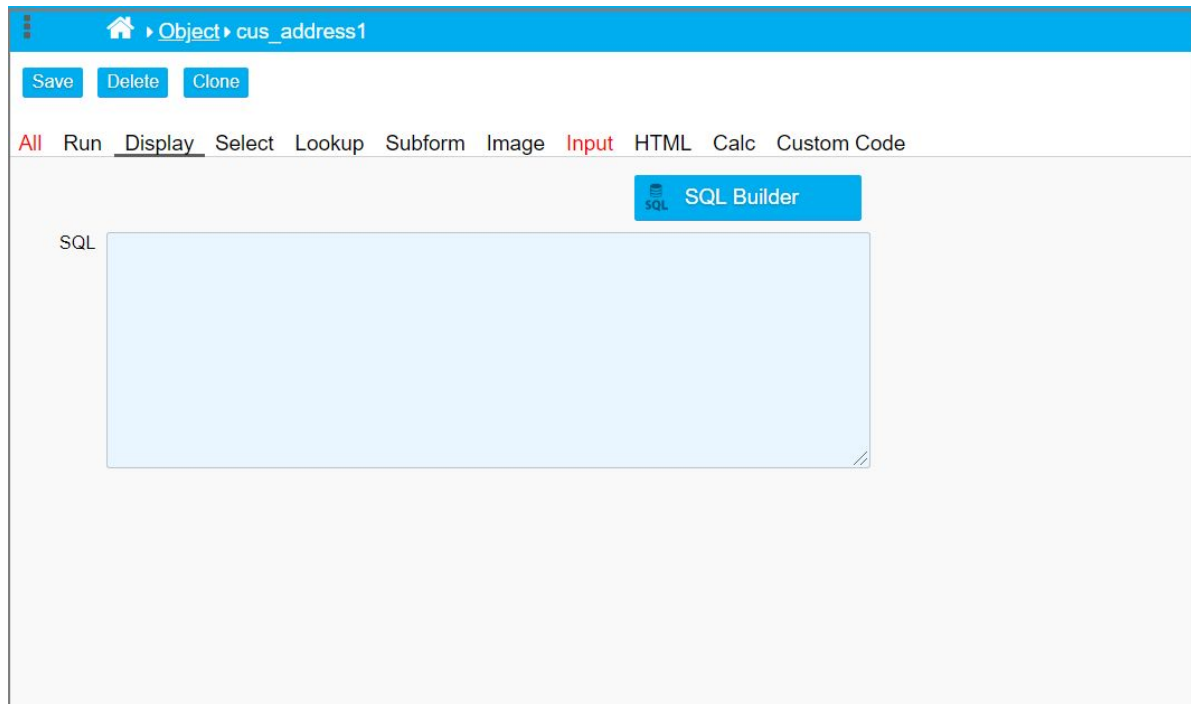
[Hash Cookies](#) can be used in this string.

Method

Choose Button or iFrame.

Button will create a button that will take you to the Form, Report or Procedure.

iFrame will mash an iFrame into the existing Edit Form that will display the Form, Report or Procedure.



Tab:Display

SQL

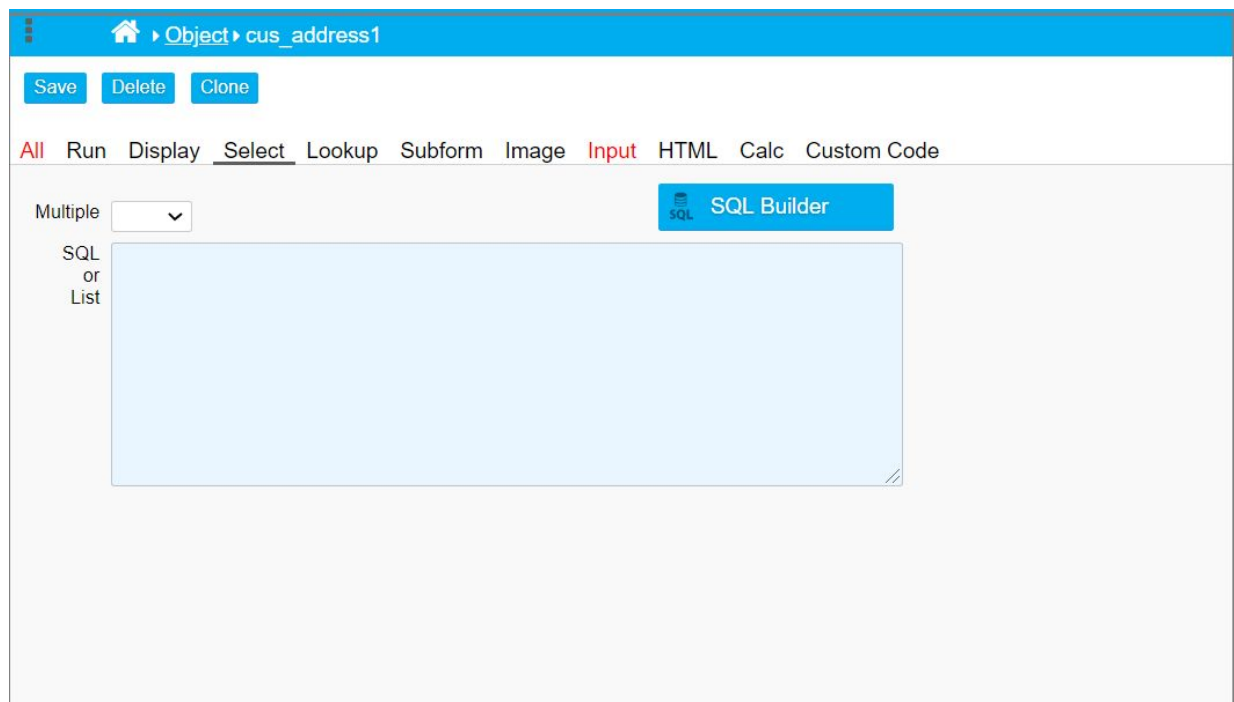
A valid SQL query that will display the first field of the first record from its result.

If you want any help building SQL, the SQL Button will direct you to nuBuilder's [SQL Builder](#).

Ash Cookies can be used within the SQL statement.

Eg.

```
SELECT cus_name FROM customer WHERE customer_id = '#RECORD_ID#'
```



Tab:Select

Multiple

Choosing Yes will allow for more than 1 selection from a visible list. Choosing No will allow for only 1 selection from a dropdown list.

SQL or List

The resulting list for this Object can be created 1 of 2 ways.

1. SQL - A valid SQL query that returns 2 columns - If you want any help building SQL, the SQL Button will direct you to nuBuilder's [SQL Builder](#).
2. List - A list delimited by a | eg. 0|No|1|Yes

Object: cus_address1

Save Delete Clone

All Run Display Select Lookup Subform Image Input HTML Calc Custom Code

Form Q

Code

Description Width

AB After Browse

Javascript

LUJS

Tab:Lookup

A Lookup Object can find and display a record selected from another Browse Form.

Form

Select a previously created nuBuilder Form.

Code

Scroll through to choose the Field to display as the *lookup able* Code from the list of possible fields.

Description

Scroll through to choose the Field to display as the Description from the list of possible fields.

Width

The width of the Description field in pixels.

After Browse

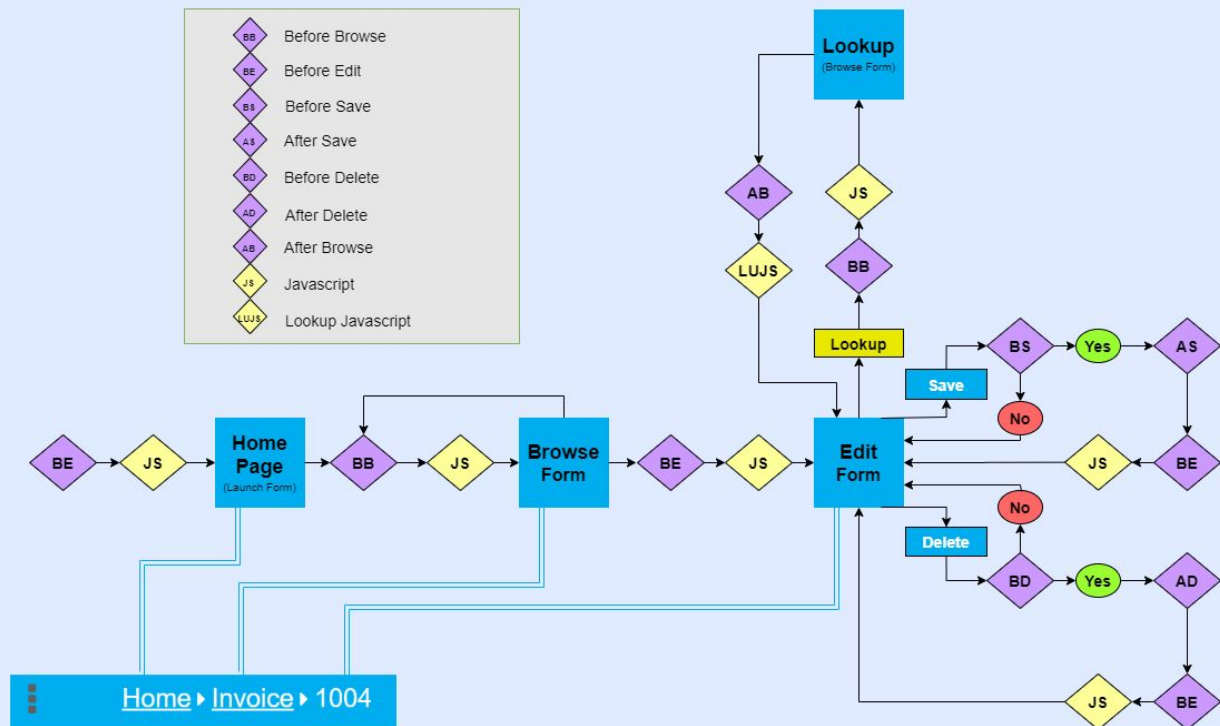
Allows [PHP](#) to be run after a record is selected.

AB in the flowchart below.

Javascript

Allows Javascript to be run after a record is selected.

LUJS in the flowchart below.



Tab:Subform

A subform is scrolling instances of another Form.

With each having a Delete checkbox on it's right - that will delete that record if ticked when the main Edit Form is saved. (if [Deleteable](#))

Invisibly linked by a [Foreign Key](#) to the main Edit Form.

This Form can appear in 2 different ways.

1. As an **Edit** Form - as normal.
2. In a **Grid** format - all Objects in a horizontal row.

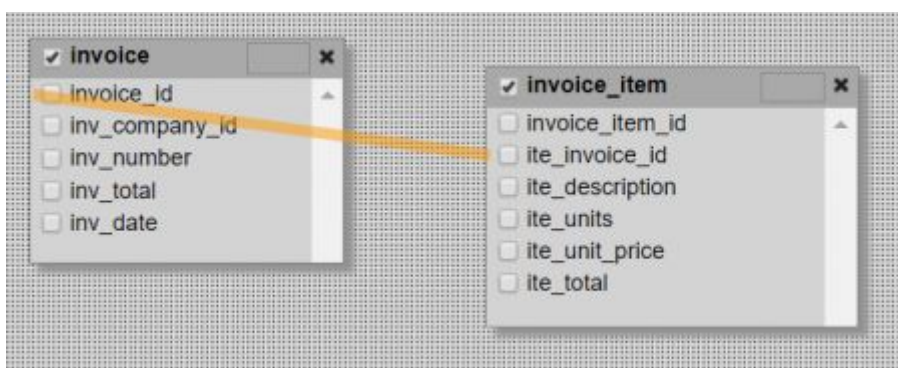
Form

Select a previously created nuBuilder Form that will scroll inside this Subform.

Foreign Key

Scroll though to choose the Foreign Key from the list of possible fields.

A Foreign Key is a field that is used to link to another table's Primary Key.



If a Subform's Form has an Object with the same ID as the Subform's Foreign Key, Subform records cannot be saved.

Addable

Select Yes or No

If Yes, there will always be a blank Form at the end of the scrolling list. So that a new record can be added.

Deletable

Select Yes or No

If Yes, each Form will have a Delete checkbox on it's right - that will delete that record if ticked, when the main Edit Form is saved.

Type

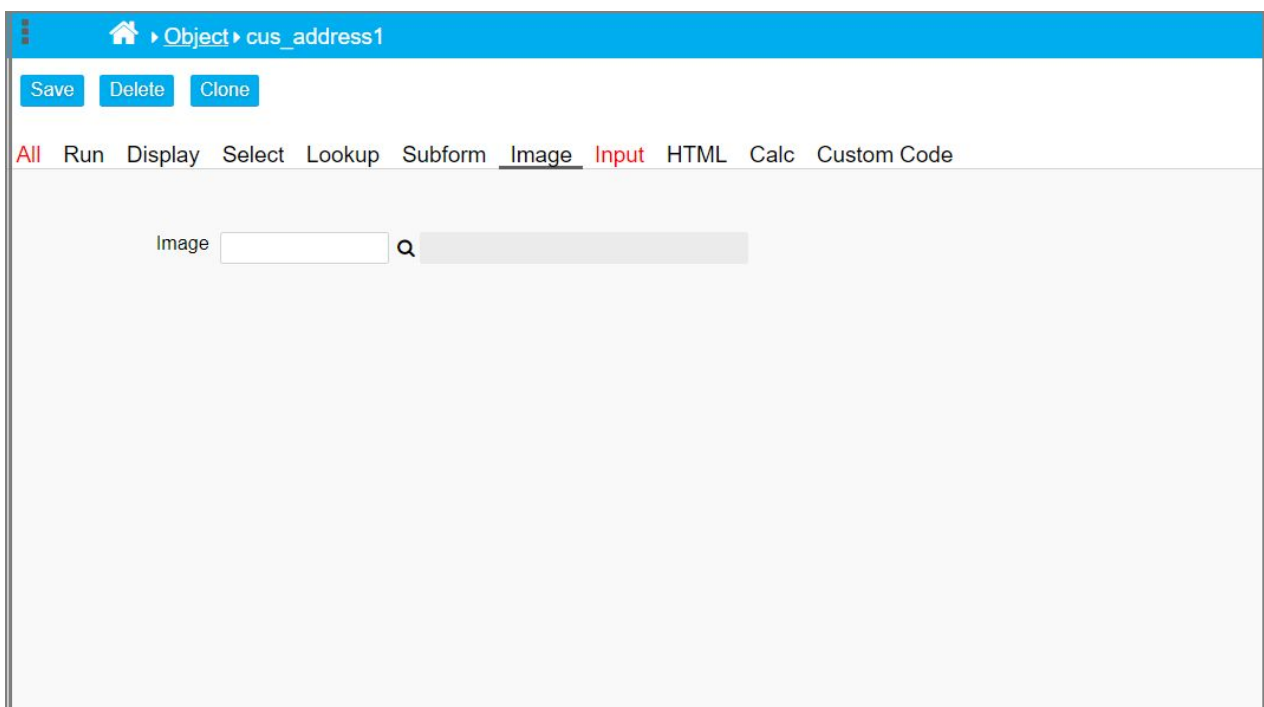
Select Grid or Form.

Inside a Subform, records are repeated either as a

- Form which displays the same as an Edit Form.

OR

- A Grid where each Object is placed in order side by side, making the height of each **Form** shorter.

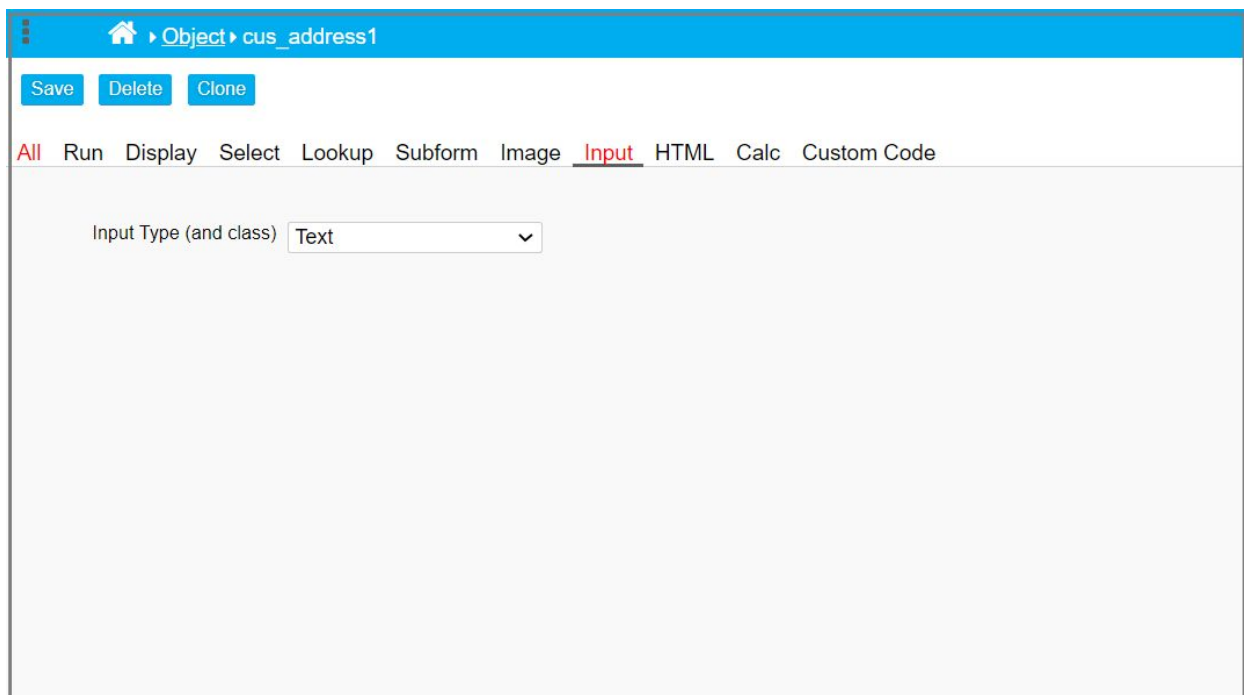


Tab:Image

The Image Object is an easy way to display a specific image stored in nuBuilder [Files](#).

Image

Choose from previously saved images.



Tab:Input

A nuBuilder Input allows for most HTML5 input types as well as 4 types just available in nuBulder Forte.

Each of these 4 Objects requires extra information and will display an extra Object when selected.

nuDate

nuDate is a nuBuilder Input field that will display a calendar as it receives focus and will format data entry or calendar selection, once edited.

It requires **Format** - A list of customisable [Formats](#).

nuAutoNumber

nuAutoNumber is a read-only nuBuilder Input field that will be populated once the Edit Form it is on, is saved for the first time.

It will be populated with an incrementing number.

It requires **Next Number** - The number this Autonumber will start from.

nuNumber

nuNumber is a nuBuilder Input field that will format data entry once edited.

It requires **Format** - A list of customisable [Formats](#).

Do not enter more than 20 characters.

nuScroll

nuScroll is a nuBuilder Input field that will allow the user to scroll through a list using keyboard up and down arrows but also allows for adding text, not in the list.

It requires a **Javascript Array**.

File

This is an HTML Object used for selecting a file to upload.

This file is stored in a JSON string and can be used for embedding files inside a div on an Edit Form using `nuEmbedObject()`.

Other HTML Types

- Button
- Checkbox
- Color
- Datetime-Local
- Email
- Hidden
- Image
- Month
- Number
- Password
- Radio
- Range
- Reset
- Search
- Telephone
- Text
- Time
- URL
- Week

a nuBuilder File Object converts a selected file to a JSON string (this is not an efficient way to handle files over 300kB).

This string can easily be used with HTML5 objects like `<EMBED>`.

The screenshot shows the Object Editor interface for an object named 'cus_address1'. At the top, there are buttons for 'Save', 'Delete', and 'Clone'. Below these is a tab bar with options: 'All', 'Run', 'Display', 'Select', 'Lookup', 'Subform', 'Image', 'Input', 'HTML' (which is selected and underlined), 'Calc', and 'Custom Code'. The main area contains configuration fields for the HTML tab: 'Chart Type' (a dropdown menu), 'Javascript Array' (a yellow input field), 'Title' (a text input field), 'Vertical Label' (a text input field), and 'Horizontal Label' (a text input field). Below these fields is a section labeled 'OR...' followed by an 'HTML' label and a large orange text area for entering custom HTML code.

Tab:HTML

This Object can be used for displaying custom HTML or a Google Chart.

Chart

Chart Type

- Pie Graph
- Line Graph
- Bar Graph
- Bar Graph - Stacked
- Bar Graph - Horizontal
- Bar Graph - Horizontal and Stacked

Javascript Array

This could be a hardcoded array,

```
[
    ['Month', 'Shane', 'Dave', 'Adam', 'Paul', 'Chris'],
    ['2004', 100, 200, 300, 400, 500],
    ['2005', 165, 238, 322, 498, 550],
    ['2006', 165, 938, 522, 998, 450],
    ['2007', 135, 1120, 599, 1268, 288]
];
```

or a Javascript function that returns an array like ...

```
nuSubformObject('items').chartData
```

Title

Appears at the top of the Chart.

Vertical Label

Appears at the left of the Chart.

Horizontal Label

Appears at the bottom of the Chart.

HTML

HTML

This will display any HTML that can be put in a DIV.

It can also contain [Hash Cookies](#) and can be used to change the HTML before it is run.

Tab:Calc

A Calc Object is a numeric field that will be recalculated after any Object's onchange event.

Format

Choose from a list of customisable [Formats](#)

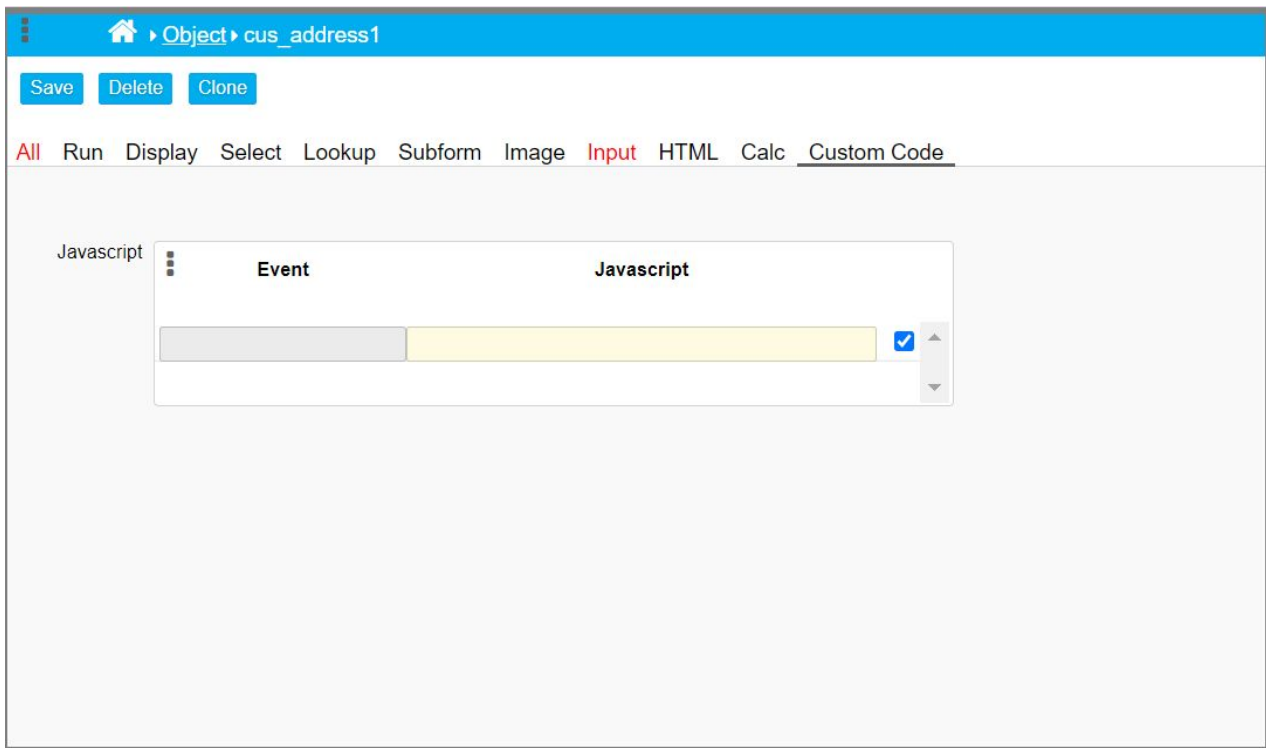
Formula

The Formula for this calculation is simply created by clicking a combination of Objects and Operators.

Objects is a list of available *Input:Number*, *Input:nuNumber*, *nuCalc* and *Select* Objects on the Edit Form.

Options containing a . refer to the Subform Name and the Input Name on that Subform. And will total up all values for that Object eg. **invoice_item_sf.ite_total**

Options not containing a . refer to an Object on the Edit Form eg.**inv_total**



Tab:Custom Code

If an Object has a Javascript event, the Custom Code Tab will be in bold.

These events are standard HTML5 events such as onclick or onchange.

The exception to this is a Subform Object which has only 3 events.

- beforeinsertrow

(Setting nuCancel = true; in this event, will stop a row being added to the Subform Object)

```
if(nuSubformObject('item_sf').rows.length > 9){  
    if(nuSubformRow == 9){  
        alert('Only 10 rows allowed');  
    }  
    nuCancel = true;  
}
```

- afterinsertrow
- clickdelete

Javascript

Event

A valid event name.

Javascript

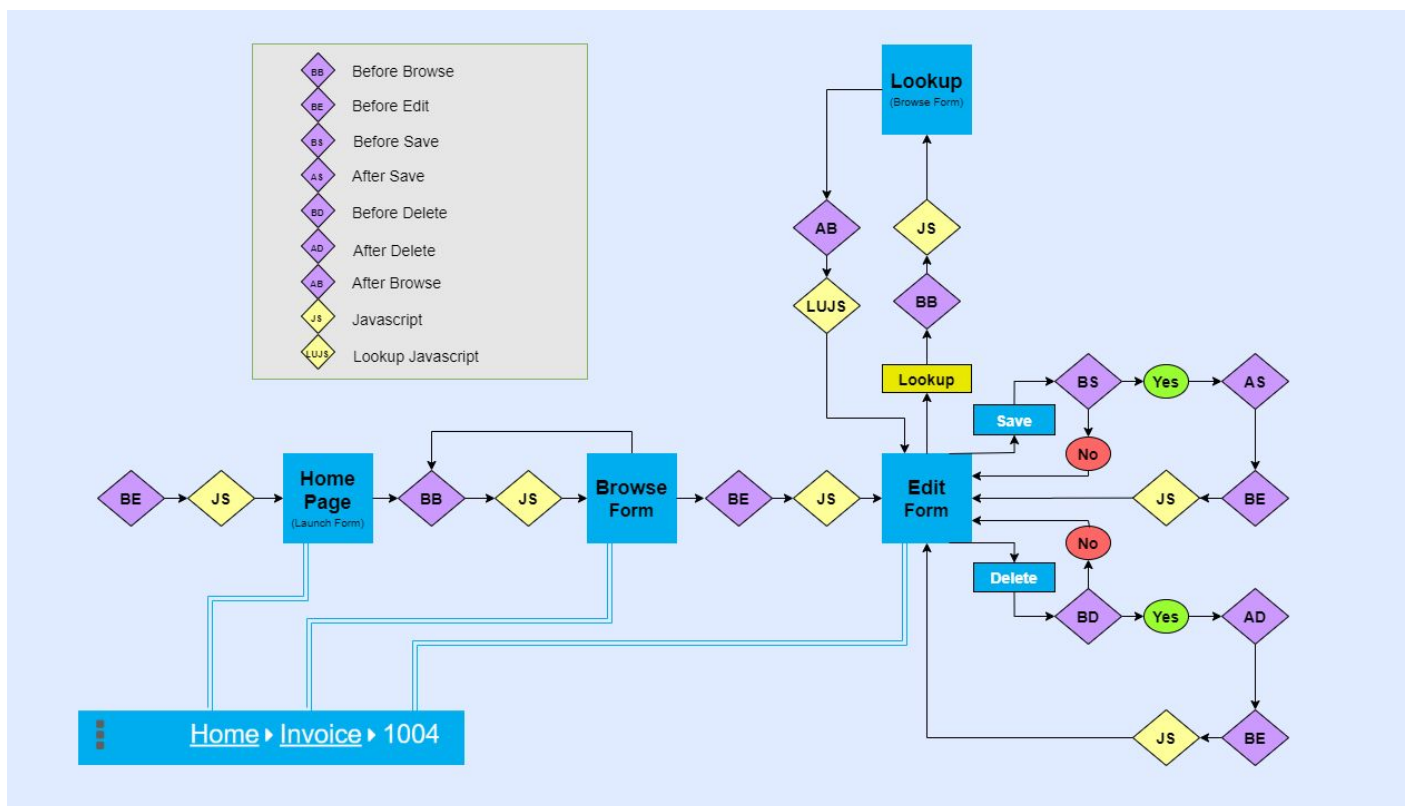
An Input Object (text field) containing Javascript code.

To edit this, double click on the Textarea and you will open Ace Editor in full screen.

PHP can be run at different event points of a Form's loading and saving.

Javascript can be used to automate certain user functions or simply change colors on a Form.

Here are the different points along nuBuilder's Flow that can be customised...



Arranging Objects

Once Objects have been placed on an Edit Form they can be Adjusted with the **Arrange Objects** Form.

The screenshot displays the 'Arrange Objects' form, which is used to adjust the layout of objects on an edit form. The form is divided into two main sections: 'Object Tab Order' and a visual representation of the objects on the form.

Object Tab Order: A list of objects is shown, with 'cus_suburb' selected. A yellow callout box with the number '1' points to this list.

Arrangement Options: A grid of buttons allows for various adjustments:

- Up, Down, Vertical, Horizontal, Shortest, Tallest, Left, Right, Thinnest, Widest
- Top, Bottom, Move to Tab, Save

A yellow callout box with the number '2' points to the 'Move to Tab' button.

Visual Representation: The right side of the form shows a visual representation of the objects on the form. The objects are arranged in a vertical stack, with 'cus_suburb' highlighted. A yellow callout box with the number '3' points to this visual representation.

Reports

The screenshot shows a web interface for building a report. At the top, there's a blue header bar with a home icon and the text 'Build Report > Build Report'. Below the header, there are three buttons: 'Save', 'Delete', and 'Clone'. The main area is titled 'Report' and contains several input fields and buttons. The 'Code' field has the value 'C'. The 'Description' field has the value 'Customer List'. The 'Group' field has the value 'Fast Report'. The 'Table' field has the value 'nuTABLE' and a search icon, with a dropdown menu showing 'customer' and an 'Open' button. The 'Launch From' field has the value 'nublack' and a search icon, with a dropdown menu showing 'Blank'. At the bottom, there is a blue button with a document icon and the text 'Report Designer'.

Code

A unique code for this Report

Description

The description of this Report

Group

A field that can be used to help organise Reports.

Table

A Table can be created in 3 different ways.

1. A single table. - (nuTABLE)
2. An SQL statement created with the [SQL Builder](#). - (nuSQL)
3. A PHP [Procedure](#).

A Procedure will only appear in this list if #TABLE_ID# is found somewhere within it.

The point of using a [Procedures](#) is to create a temporary table that can be manipulated before it is used by the Report.

(If a nuTABLE or a nuSQL *table* is chosen, this temporary table is created automatically)

The temp table created must be called **#TABLE_ID#**.

#TABLE_ID# simply gets replaced by the name of a temporary table, created by nuBuilder, that will be used by the Report.

It works the same way as other Hash Cookies.

```
$s = "

    SELECT
        sta_last_name,
        sta_initials,
        sta_games,
        sta_runs,
        sta_captain,
        CONCAT(sta_last_name, ',', sta_initials) AS full_name,
        1 AS player

    FROM
        stats

",
'

nuRunQuery("CREATE TABLE #TABLE_ID# $s");
```

Launch From

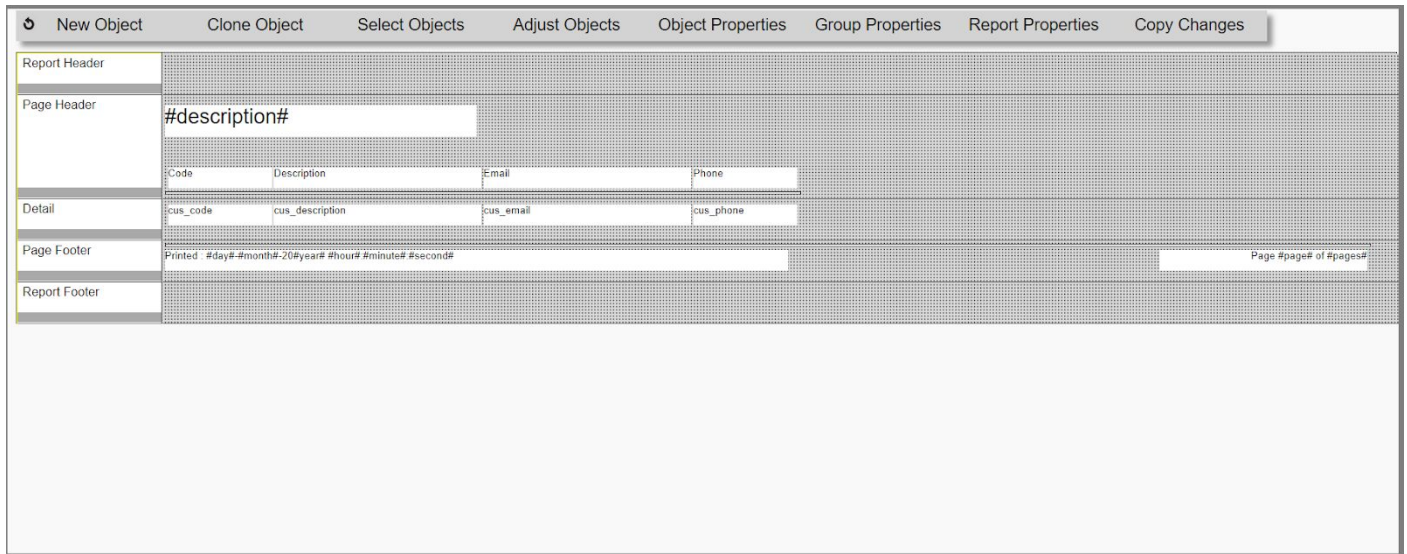
A Launch Form will allow..

- The user defines the information to be stored in the table that will be used by the report.
- Run the Report from an [Action Button](#) at the top of the [Form](#).

Report Designer

nuBuilder Forte has its own Report Designer.

Arranging Objects



The nuBuilder Forte Report Designer will open up in a new browser window.

It is a graphical user interface, with drag and drop features, for designing a [Report](#).

9 Functions

New Object

This will add a default Object into the Detail Section of the Report.

Clone Object

Doing this will clone any highlighted Objects.

Cloning an Object often saves more time than just adding an Object.

Select Objects

There are 3 ways to select one or more Objects.

1. Clicking on an Object. Holding the Ctrl Key down will allow you to add another Object to those already selected.
2. Drawing a rectangle around them with your mouse.
3. Clicking **Select Objects** on the Menu.

Adjust Object

This will bring up a dialog box with options that can adjust all selected Objects..

- Align To Left
- Align To Right
- Align To Top
- Align To Bottom
- Space Vertically
- Space Horizontally
- Thinnest
- Widest
- Tallest

- Shortest

Object Properties

- ID
- Object Type
 - Field - Displays a value that comes from the reports [Table](#).
 - Label - Displays Text - these can use [Hash Cookies](#).
 - Image - Displays a PNG Image.
- Left
- Top
- Height
- Width
- Background Color
- Border Color
- Border Width
- Field Name/Title/Source - (*Field/Label/Image*)
 - Field Name - A [Scrollable](#) list that displays a list of values that comes from the report's [Table](#).
 - Title - Text that can include [Hash Cookies](#).
 - Source - A [Scrollable](#) list that displays a list of values that comes from..
 - Field - A Fieldname in the [Table](#).
 - Image - A list of nuBuilder Forte [File](#).
 - Label - A List with only 1 choice (that can be over written with an actual Label) which is used to keep the Label to the exact size as its Height Property - KEEP EXACT HEIGHT.
- Font Color
- Font Family
- Font Size
- Font Weight
- Format
- Text Align
- Minimum Rows - In a growable Field Object.
- Maximum Rows - In a growable Field Object.
- Z Index - Objects with larger numbers will appear on top.

Group Properties

This will bring up a dialog box that will allow us to do 2 things.

1. Sort the records into a certain order before printing. - Sorting on up to 8 [Scrollable](#) fields.
2. Format each sections of the Report. - The properties for each Section are..
 - Section Height - In pixels.
 - Section Color - HTML.
 - Section Page Break - Start a new page when completed.
 - Yes.
 - No.

Report Properties

This will bring up a dialog box with 2 options.

1. Paper
2. Orientation

Copy Changes

This will copy the design back to the **Build Report** Form and close the **Report Designer**.

These changes will **NOT** be saved until the current **Build Report** record is saved!

Functions

There are 2 functions that can be used within Group and Report Headers and Footers.

1. sum() - eg. sum(inv_total)
2. average() - eg. average(inv_total, units)

For these functions to work there must be at least 1 Report Group.

Procedures

Build Procedure > HW

Save Delete Clone

PHP

Code HW

Description Greeting

Group demo

Launch from nublack Q Blank

Run In a new window

PHP print 'hello world';

nuBuilder Forte can create Procedures.

A Procedure is PHP code run to manipulate the database.

Procedures are written in PHP to do things to the database directly.

Here are 2 examples of where a Procedure can be used.

1. To create the [table](#) used in a Report.
2. Update a stock table after an invoice is created in nuBuilder.

Code

A unique code for this Procedure.

Description

The description of this Procedure.

Group

A field that can be used to help order Procedures in a list.

Launch From

This is only necessary if you need it to...

- Run it from the Run Procedure Button.
- Reference Hash Cookies from a certain Edit Form.

It can still be used by nuProcedure().

A Launch Form will allow...

- The user defines the information to be used by the Procedure.
- Run the Procedure from an Action Button at the top of the Form.

Run

- In a new window - Appears in a new window much the same as a Report.
- Hidden - This runs behind the scenes but can display a result using nuDisplayMessage

PHP

A Textarea containing PHP code.

This code can also use Hash Cookies that will help you customise nuBuilder.

To edit this, double click on the Textarea and you will open Ace Editor in full screen.

Hash Cookies

- A cookie helps hold values between Javascript and PHP.
- A nuBuilder cookie does a similar thing and has hashes around it eg. **#USER_ID#**.

There are a number of places Hash Cookies can be used...

In each of these situations, nuBuilder does a search and replace with all currently available Hash Cookies and any you use in your code before it is run.

There are 5 Hash Cookies always available.

1. USER_ID
2. USER_GROUP_ID
3. HOME_ID
4. GLOBAL_ACCESS
5. ACCESS_LEVEL_CODE

To find what other Hash Cookies are available for use, you can run [nuHash\(\)](#) inside your code.

PHP

Hash Cookies allow you to customise your [PHP code](#).

Example of using Hash Cookies in PHP.

```
$s = "CREATE TABLE #TABLE_ID# SELECT * FROM invoice";  
nuRunQuery($s);
```

HTML

Hash Cookies allow you to customise your HTML [Objects](#).

The Hash Cookies available to HTML Objects are the same as those available on **Before Edit**.

Example of using Hash Cookies in HTML.

```
<iframe  
width="560"  
height="315"  
src="#cus_youtube#"  
frameborder="0" allowfullscreen>
```

</iframe>

Custom SQL

All Custom SQL used in nuBuilder Forte.

```
CREATE TABLE #TABLE_ID# SELECT * FROM customer;
```

or

```
<source lang="sql">
```

```
SELECT * FROM customer WHERE cus_group = '#sales_group#';
```

```
</source>
```

SQL Builder

Any **Clause** in the **Clauses** Subform in the [SQL Builder](#).

```
= '#RECORD_ID#'
```

Report Labels

Using Hash Cookies allows you to have changing titles on [Report Labels](#).

An example of using Hash Cookies in a Report Label. <source lang="html">

```
Sales Between '#from_date#' And '#to_date#'
```

```
</source>
```

The following are specific hash variables available only for Report Labels are...

- page
- pages
- description
- code
- year
- month
- day
- hour
- minute
- second

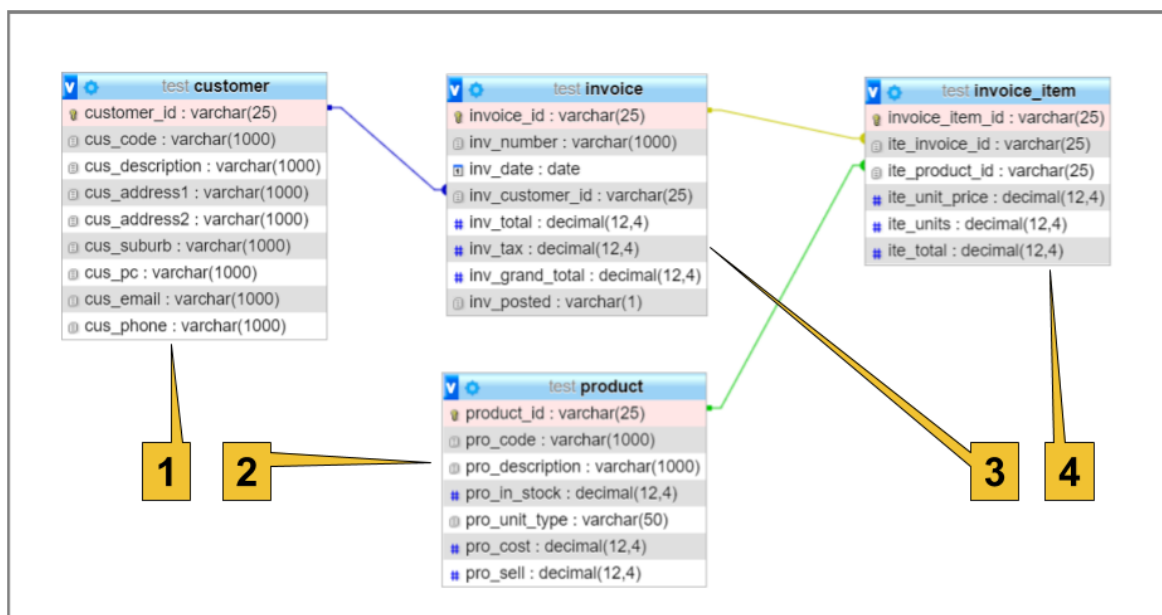
Creating an Invoicing Application

The database created by following this tutorial can be downloaded at [Source Forge](#).

This simple application could be used to take phone orders for restaurants or retail stores etc.

It will store the following information in the database.

1. Customer
2. Product
3. Invoice
4. Invoice items



This information will be entered by users through a nuBuilder Form (Invoice Form) that looks like this..

nuBuilder Invoice Form Screenshot:

- 1** Points to the **Customer** field (HPM).
- 2** Points to the **Items** table.
- 3** Points to the **Invoice No.** field (1005).
- 4** Points to the **Total** field (97.90).

Items	Product	Unit \$	Units	Total
HS	Hand Sanitizer - 300ml	5.90	11.00	64.90
TP	Roll of Toilet Paper	3.00	11.00	33.00
				0

Total: 97.90
Tax: 9.79
Grand Total: 107.69

Create Forms and Tables

To create the required Forms and tables we will build 4 Fast Forms.

- 3 **Edit and Browse** Forms
- 1 **Subform**.

1.Customer

Form Builder

Build Fast Form

Form Type: Browse and Edit

Table Name: customer

Addable Objects

Input:Text	Word
Input:Button	Calc
Input:Checkbox	Display
Input:nuAutoNumber	Lookup
Input:nuScroll	HTML
Input:nuDate	Select
Input:nuNumber	Select:Multiselect
Input:Number	Textarea
Input:File	Image
Subform	

New Fast Form

Label	Field Name	Browse Column
Code	cus_code	<input checked="" type="checkbox"/>
Description	cus_description	<input checked="" type="checkbox"/>
Email	cus_email	<input checked="" type="checkbox"/>
Phone	cus_phone	<input checked="" type="checkbox"/>
Address 1	cus_address1	<input type="checkbox"/>
Address 2	cus_address2	<input type="checkbox"/>
Suburb	cus_suburb	<input checked="" type="checkbox"/>
Postcode	cus_pc	<input type="checkbox"/>

2.Product

Form Builder

Build Fast Form

Form Type: Browse and Edit

Table Name: product

Addable Objects

Input:Text	Word
Input:Button	Calc
Input:Checkbox	Display
Input:nuAutoNumber	Lookup
Input:nuScroll	HTML
Input:nuDate	Select
Input:nuNumber	Select:Multiselect
Input:Number	Textarea
Input:File	Image
Subform	

New Fast Form

Label	Field Name	Browse Column
Code	pro_code	<input checked="" type="checkbox"/>
Description	pro_description	<input checked="" type="checkbox"/>
In Stock	pro_in_stock	<input checked="" type="checkbox"/>
Unit Type	pro_unit_type	<input checked="" type="checkbox"/>
Cost	pro_cost	<input type="checkbox"/>
Sell	pro_sell	<input type="checkbox"/>

3.Invoice

Form Builder

Build Fast Form

Form Type Browse and Edit ▼
Table Name

Addable Objects

Input:Text

Word

Input:Button

Calc

Input:Checkbox

Display

Input:nuAutoNumber

Lookup

Input:nuScroll

HTML

Input:nuDate

Select

Input:nuNumber

Select:Multiselect

Input:Number

Textarea

Input:File

Image

Subform

New Fast Form

Label	Field Name	Browse Column
Invoice No.	inv_number	<input checked="" type="checkbox"/>
Date	inv_date	<input checked="" type="checkbox"/>
Customer	inv_customer_id	<input checked="" type="checkbox"/>
Total	inv_total	<input type="checkbox"/>
Tax	inv_tax	<input type="checkbox"/>
Grand Total	inv_grand_total	<input checked="" type="checkbox"/>
Posted	inv_posted	<input checked="" type="checkbox"/>
Items	invoice_item_sf	<input type="checkbox"/>
		<input checked="" type="checkbox"/>

4.Invoice Item

Form Builder

Build Fast Form

Form Type Subform ▼
Table Name
Foreign Key Field Name

Addable Objects

Input:Text

Word

Input:Button

Calc

Input:Checkbox

Display

Input:nuAutoNumber

Lookup

Input:nuScroll

HTML

Input:nuDate

Select

Input:nuNumber

Select:Multiselect

Input:Number

Textarea

Input:File

Image

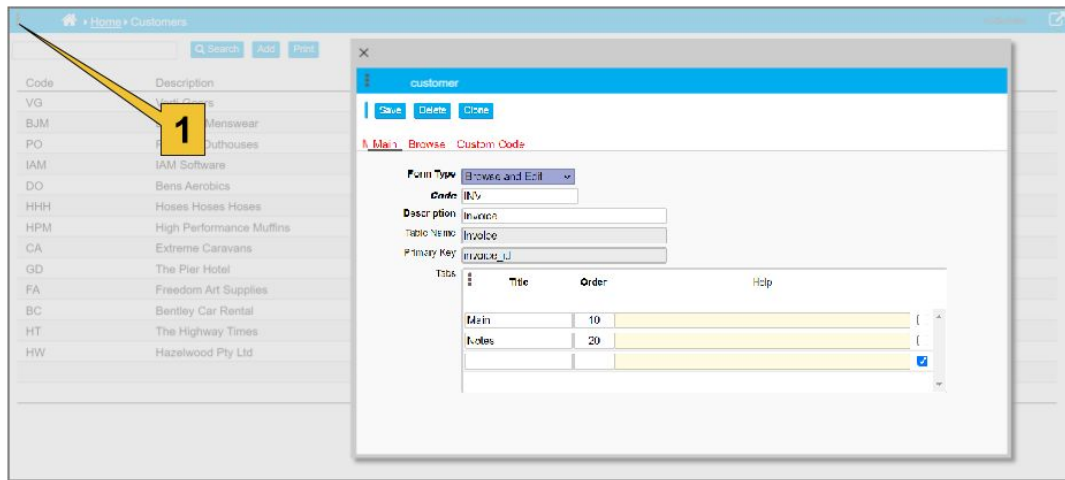
Subform

New Fast Form

Label	Field Name	Browse Column
Product	ite_product_id	<input type="checkbox"/>
Unit \$	ite_unit_price	<input type="checkbox"/>
Units	ite_units	<input type="checkbox"/>
Total	ite_total	<input type="checkbox"/>
		<input checked="" type="checkbox"/>

Edit Form Properties

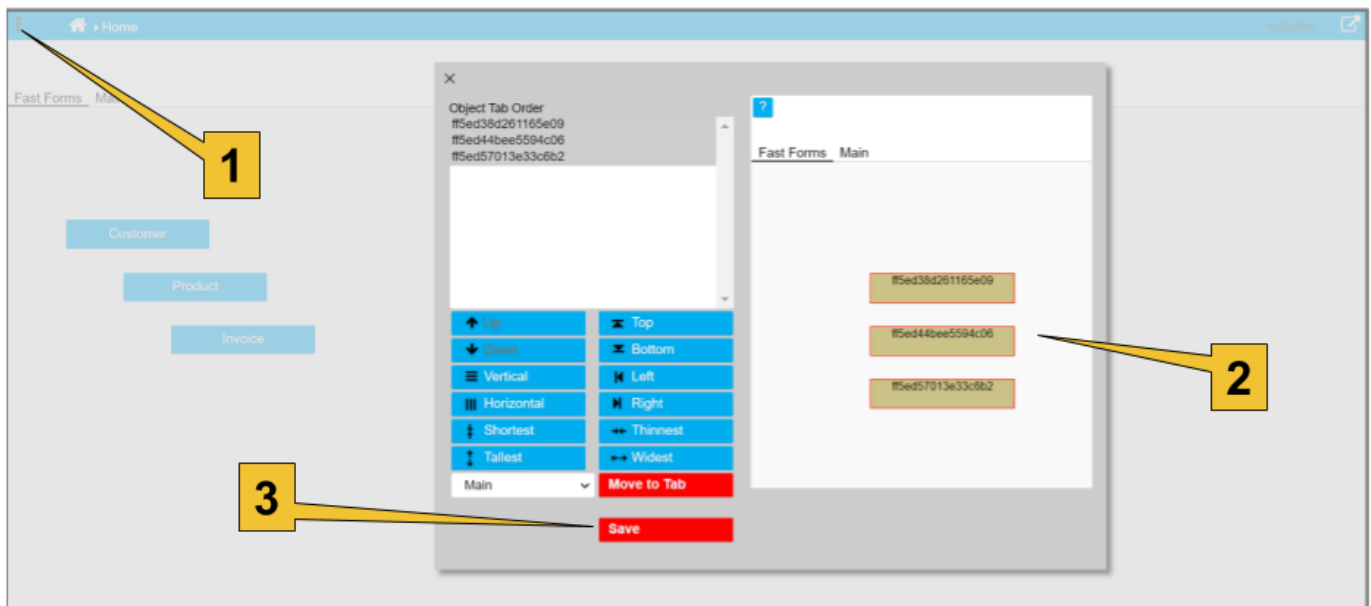
Forms created with Fast Forms will require the default Code and Description to something more descriptive.



1. Click the Options icon in the top left corner of a Form or Subform and select **Form Properties**..

Arrange Objects

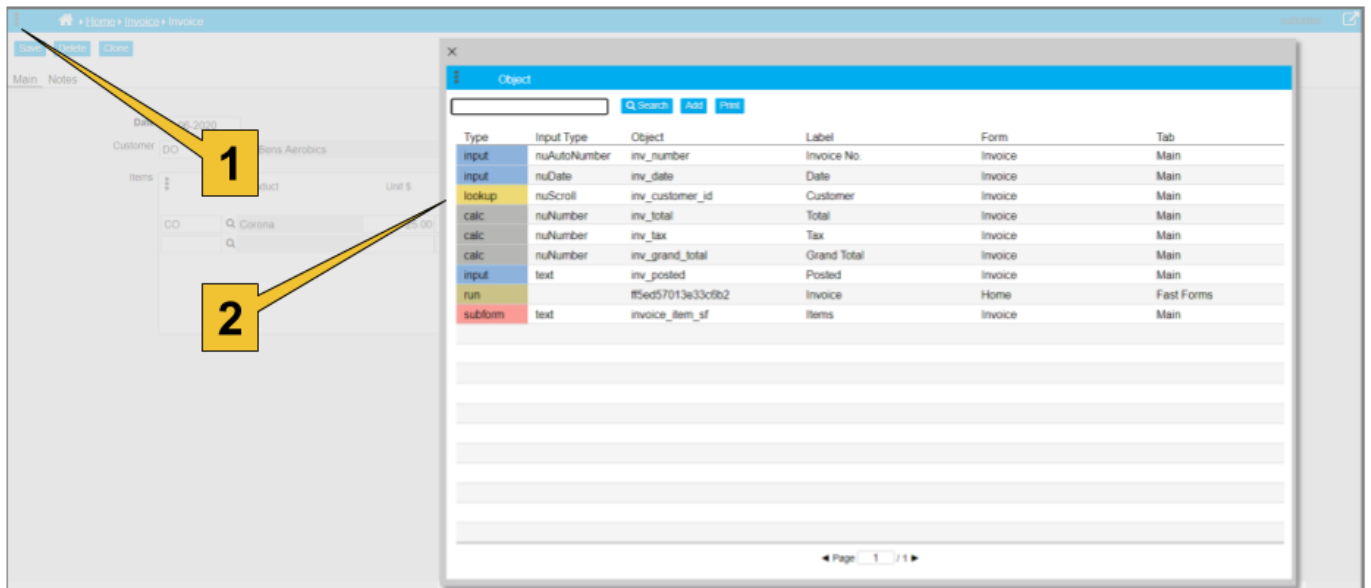
A Fast Forms builds a Form and adds Objects. These Objects will need adjusting so they appear on an Edit Form in a tidy logical order. A Fast Form may also create a Button to open the new Form, on the User Home Page (which will also require adjusting) eg.



2. Click the Options icon in the top left corner of an Edit Form or Subform.
3. Highlight the Objects you want to adjust.
4. Click Save to apply these changes to that Form.

Edit Object Properties

Some objects created with a Fast Form have default values that will need editing.



There are 3 ways to view and edit an Object's properties.

1. Via the Options icon (as above).
 - 1.1. Select **Form Object List**
 - 1.2. Select Object.
2. By double clicking and Object's Label if it has one.
3. Via the **Object** Form on the Setup Tab.

Edit These Objects.

- inv_customer_id

On the Lookup Tab.

- Set **Form** to Product Form.
- Set **Code** to pro_code.
- Set **Description** to pro_description.

- ite_product_id

On the Lookup Tab.

- Set **Form** to Product Form.
- Set **Code** to pro_code.
- Set **Description** to pro_description.
- Add the following to **After Browse** to update the item's unit price once a product is selected...

```
$lu = nuLookupRecord();  
nuSetFormValue('ite_unit_price', $lu->pro_sell);
```

- ite_total - Set **Formula** to `nuTotal('ite_unit_price') * nuTotal('ite_units')`
- inv_total - Set **Formula** to `nuTotal('invoice_item_sf.ite_total')`
- Inv_tax - Set **Formula** to `nuTotal('invoice_item_sf.ite_total') * .1`
- Inv_grand_total - Set **Formula** to `nuTotal('invoice_item_sf.ite_total') * 1.1`

For more information on an Object's properties refer to [Objects](#).

Create an Invoice Docket

Next, we create a delivery docket that itemises the customer, product and pricing information that can be run from the Invoice Edit Form.

Build SQL

The screenshot shows the SQL Builder interface with the following components:

- Buttons:** Save, Delete, Clone.
- SQL Section:**
 - Description:** invoice
 - Add Table:** dropdown menu
 - Resize:** button
- Fields and Relationships:**
 - customer:** ☒ customer_id, ☐ cus_code, ☐ cus_description, ☐ cus_address1, ☐ cus_address2, ☐ cus_suburb
 - invoice:** ☐ invoice_id, ☒ inv_number, ☒ inv_date, ☐ inv_customer_id, ☒ inv_total, ☒ inv_tax, ☒ inv_grand_total
 - invoice_item:** ☒ invoice_item_id, ☐ ite_invoice_id, ☐ ite_product_id, ☐ ite_unit_price, ☐ ite_units, ☐ ite_total
 - product:** ☐ product_id, ☐ pro_code, ☒ pro_description
- Clauses:**

Type	Field	Clause	Sort	Order
WHERE	invoice.invoice_id	= '#USER_ID#'		10
- SQL:** From SQL Builder
- SQL Query:**

```
SELECT
customer.*,
invoice.inv_number,
invoice.inv_date,
invoice.inv_total,
invoice.inv_tax,
invoice.inv_grand_total,
invoice_item.*,
product.pro_description
```

```
SELECT
customer.*,
invoice.inv_number,
invoice.inv_date,
invoice.inv_total,
invoice.inv_tax,
invoice.inv_grand_total,
invoice_item.*,
product.pro_description

FROM
customer
JOIN invoice ON customer.customer_id = invoice.inv_customer_id
JOIN invoice_item ON invoice.invoice_id = invoice_item.ite_invoice_id
JOIN product ON product.product_id = invoice_item.ite_product_id

WHERE
((invoice.invoice_id = '#RECORD_ID#'))
```

Create Fast Report

Fast Report

Build Fast Report

Main

Table Data

nuSQL

Q invoice

Order By

customer_id

Available Fields

customer_id

cus_code

cus_description

cus_address1

cus_address2

cus_suburb

cus_nc

A Fast Report has been created!

(There is now a Report with a Code of **FR1** found in **Reports**)

New Fast Report

Title	Field Name	Width	Sum
cus_phone	cus_phone	100	<input type="checkbox"/>
inv_number	inv_number	100	<input type="checkbox"/>
inv_date	inv_date	100	<input type="checkbox"/>
inv_grand_total	inv_grand_total	100	<input type="checkbox"/>
pro_description	pro_description	100	<input type="checkbox"/>
ite_unit_price	ite_unit_price	70	<input type="checkbox"/>
ite_units	ite_units	70	<input type="checkbox"/>
ite_total	ite_total	70	<input type="checkbox"/>
			<input checked="" type="checkbox"/>

Used

1110

Remaining

40

Edit Report

Build Report

Build Report

Save

Delete

Clone

Report

Code

FR1

Description

Fast Report 1

Group

Fast Report

Table

nuSQL

Q invoice

Launch From

nublack

Q Blank

Open

Report Designer

Rename Code and Description.

Adjust Report Design from...

	New Object	Clone Object	Select Objects	Adjust Objects	Object Properties	Group Properties	Report Properties	Copy Changes
Report Header								
Page Header								
Detail								
Page Footer								
Report Footer								

To...

	New Object	Clone Object	Select Objects	Adjust Objects	Object Properties	Group Properties	Report Properties	Copy Changes
Report Header								
Page Header								
Detail								
Page Footer								
Report Footer								

To print this...

Docket No. : 1005		
Date : 2020-06-12		
High Performance Muffins		
65 Tree Tce		
Jamestown		
080 345 6745		
Description	Units	Total
Roll of Toilet Paper	11.00	33.00
Hand Sanitizer - 300ml	11.00	64.90
Total :		107.69

Printed : 20-06-2020 11:19:52

Add Action Buttons

We can add an Action Button to the Invoice Form that will Print the current docket.

This is done by adding Javascript to the invoice Form

We will add 2 Action Buttons to the invoice Edit Form.

- A Posted Button - This will set a flag (inv_posted) to 1. An invoice that is not posted will not see the Print Button.
- The Print Button - .This will appear only when an invoice is posted.

```
if(nuFormType() == 'edit'){  
    if($('#inv_posted').val() == '1'){  
        nuAddActionButton('pinv', 'Print Docket', 'nuRunReport("INV");'  
    }else{  
        nuAddActionButton('ppost', 'Post Docket', '$("#inv_posted").val(1).change();nuSaveAction();'  
    }  
}
```

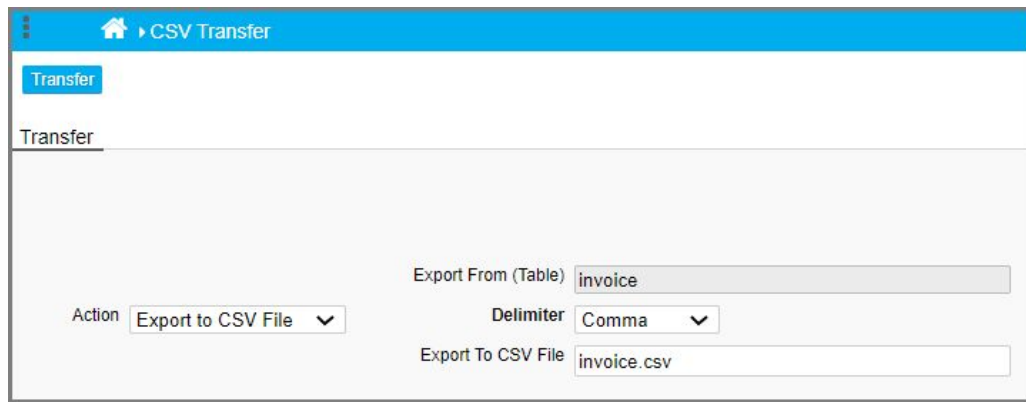
CSV Transfer

The CSV Transfer Form Allows...

Exporting a database table to a CSV file.

Importing CSV files to a new database table.

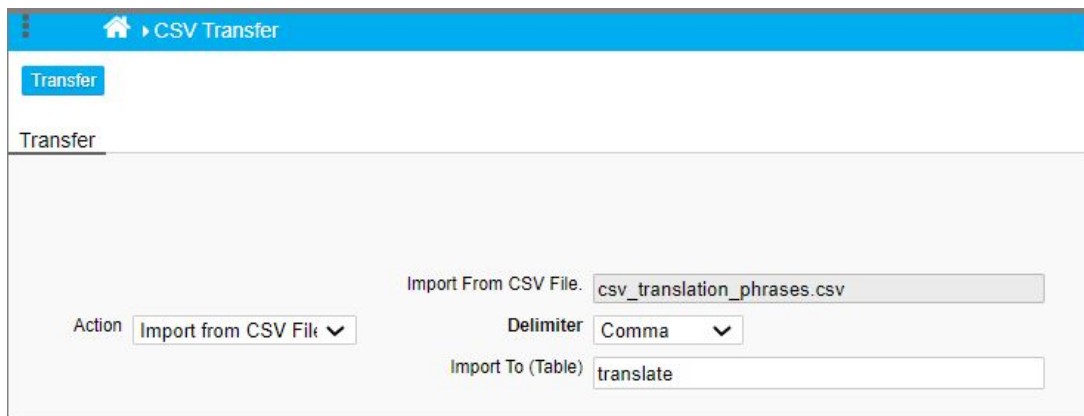
Exporting a CSV File



The screenshot shows the 'CSV Transfer' interface. At the top, there is a blue header bar with a home icon and the text 'CSV Transfer'. Below the header, there is a 'Transfer' button and a 'Transfer' tab. The main content area is a light gray rectangle. At the bottom, there are three input fields: 'Action' with a dropdown menu set to 'Export to CSV File', 'Export From (Table)' with a text box containing 'invoice', and 'Export To CSV File' with a text box containing 'invoice.csv'. A 'Delimiter' dropdown menu is also present, set to 'Comma'.

1. Export to CSV File. Choose the CSV file from the nuScroll list.
2. Delimiter. Choose the Delimiter to space each field value.
3. Import to (CSV). Enter a filename for the new CSV file.
4. Click the Transfer Action Button.

Importing a CSV File



The screenshot shows the 'CSV Transfer' interface. At the top, there is a blue header bar with a home icon and the text 'CSV Transfer'. Below the header, there is a 'Transfer' button and a 'Transfer' tab. The main content area is a light gray rectangle. At the bottom, there are three input fields: 'Action' with a dropdown menu set to 'Import from CSV File', 'Import From CSV File' with a text box containing 'csv_translation_phrases.csv', and 'Import To (Table)' with a text box containing 'translate'. A 'Delimiter' dropdown menu is also present, set to 'Comma'.

1. Import from CSV File. Choose the database table from the nuScroll list.

To appear in the nuScroll List the CSV File must be placed in the "csvfiles" subdirectory inside nuBuilder Forte's directory - on the server.

2. Delimiter. Choose the Delimiter to space each field value.
3. Import to (Table). Enter a filename for the new CSV file.

4. Click the Transfer Action Button.