

# SailformsPro Application Development Guide

Version 2.0



**SailformsPro** is a small business database, developed to make it easy to build custom entry forms. Reports are automatic, easy to sort, filter, total and subtotal for quick analysis and export.

Single Form databases work fine for simple data. Your business data requires Forms to interact with other Forms. A Relational Database allows two or more Forms to connect and interact with each other.

This guide will take you through all of the functions available in **SailformsPro**. It assumes you have some knowledge of setting up **Sailforms'** Forms.

## New to **SailformsPro**

- Entry Fields - **Active Connections**
  - Search-List-Get
  - Get Value from
  - Put Value to
  - Add to
  - Subtract from
- Remotes/Controls
  - Remote Table
  - Remote Field
  - Search Remote
  - Total Remote
  - Min/Max Remote
  - Count Remote
  - Buttons
- Multiple Form Layouts
  - Fixed layout Landscape or Portrait or Both
  - Specific Layouts for different Screen sizes
  - Button Command selectable alternate layouts

Let's look at a common multiple Form problem, the Invoice.

You can load the forms used in this tutorial from the Pro System Samples, **Invoice Tutorial**, from the Main menu in **Sailforms**.

## Invoicing

An Invoice is the main entry in a Form containing multiple entries of line items, sales of items to a customer.

We want to be able to:

- Auto number the Invoice
- Pick a customer
- Enter items sold and keep track of inventory
- Total the order and add tax

### Create a new Group



From the **Sailforms** Home screen, press Menu. ☰

Select **New Group**.



Enter a Group Name *Inventory Control*.

Press ✓ to save the new Group.

You can add an optional password to limit access to this group.

You can add and remove the password at any time.



Scroll to the bottom and select the new **Inventory Control** Group by tapping on it.

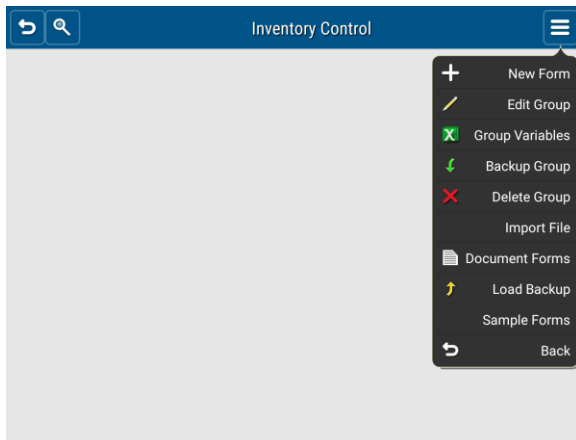
## What you sell

Now we need to make the form to hold items or services that we want to sell.


This form is used to keep track of your Items.

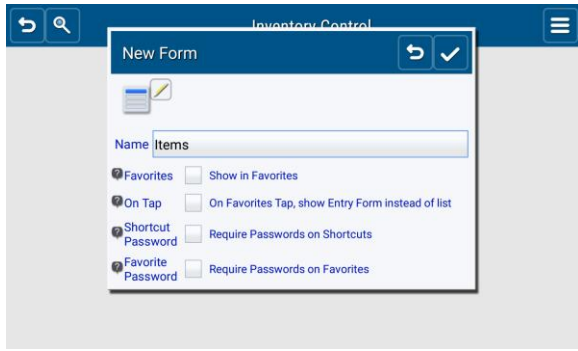
## Inventory Items Fields

- Item Id with Barcode reader
- Item Name
- Quantity on Hand
- Quantity Sold
- Price



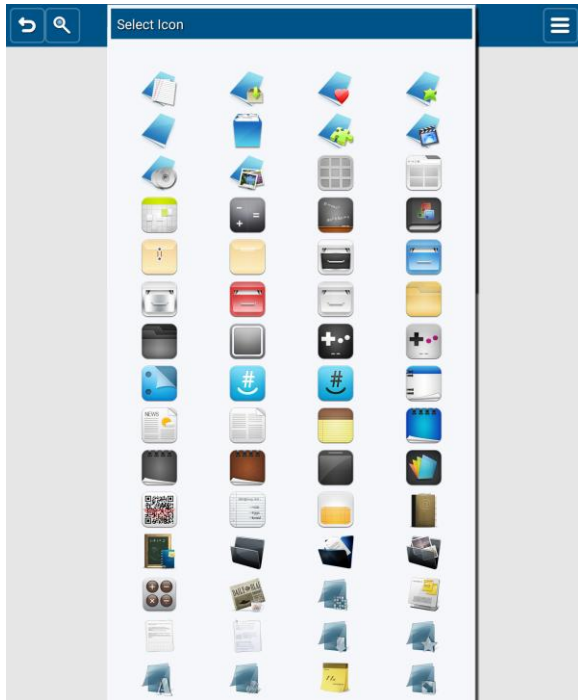
The new Group has no forms.

Press Menu  and select **New Form**.

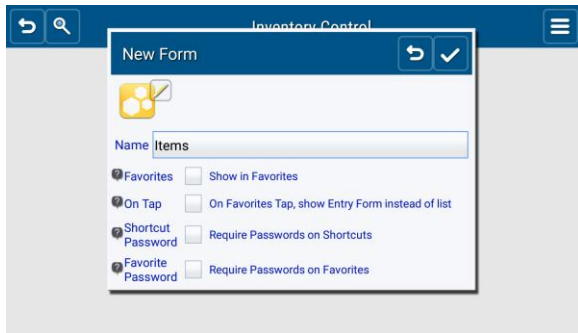


Enter the new form name *Items*.

Tap the Icon  to Pick an Icon.



Find an Icon and tap it to select it.

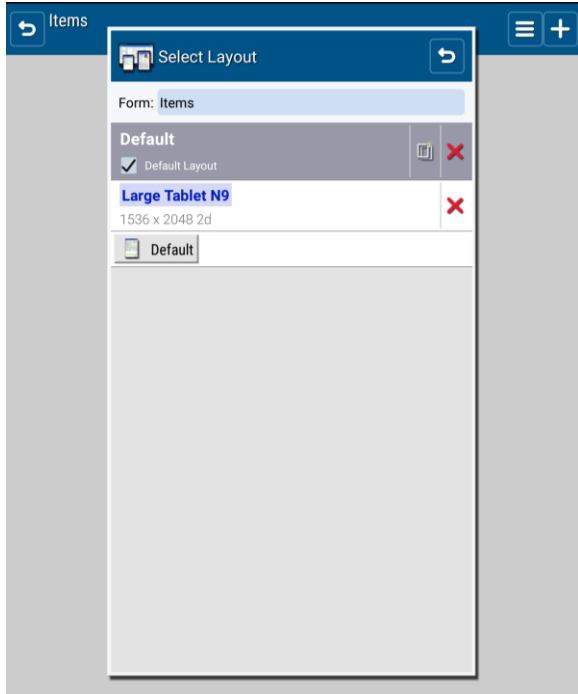


Press  to save the new Form.




## Multiple Layout Support

**SailformsPro** allows you to setup multiple layouts per form. Each layout can support multiple screen sizes for any of the Android devices.



You can also define alternative layouts that can be displayed by Button Commands to create Work Flows for your forms.

For now, just pick the single **Default** layout. 

**Sailforms** has predefined screen sizes for many of the common devices. If your device is not found, it will create a screen size to match your device.

Each form can have more than one screen size layout per form.

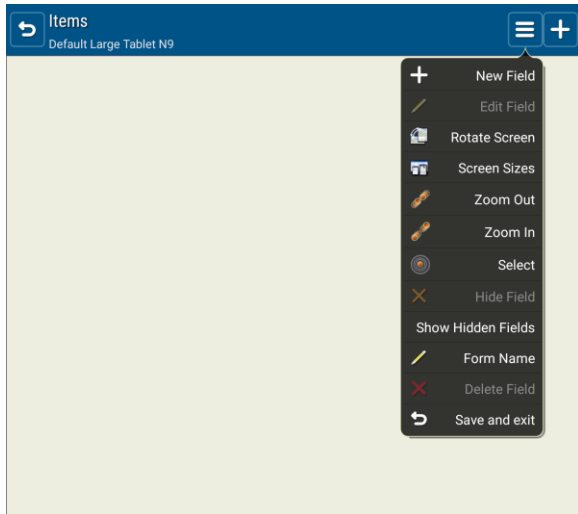
You can have a fixed layout for both Portrait and Landscape and you can

limit the layout to one or the other.

You can create a layout for each device screen size. **Sailforms** will pick the best size when it runs. See the section on how **Sailforms** Picks layouts for more information.

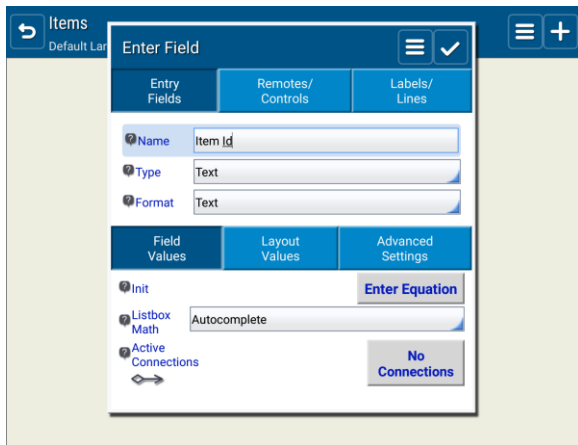
### Alternate layouts for a form.

You can also create a set of layouts for a form that can be controlled using button commands. See the section on **Alternate layouts** for more information.



Add a new field, let's start with Item Id.

Press  to create a new Field.

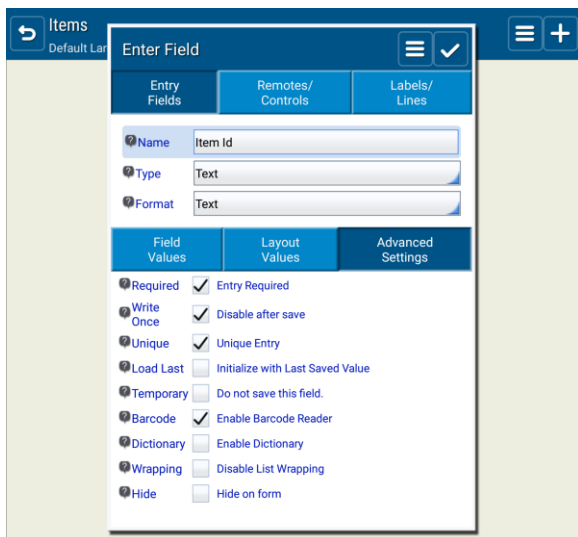


Enter the name, *Item Id*.

You can set the Type to Text or Numbers for your Item Ids.

Tap the **Advanced Settings** tab.

The Item Id will be a **Unique Key** to lookup and create an **Active Connection**.




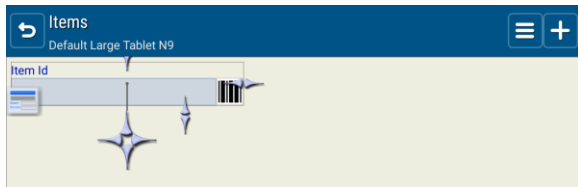
Since this is a Unique Key, check the check boxes, **Required**, **Write Once** and **Unique**. These will force this to always be a Unique Key.


Check the **Barcode** checkbox to allow you to look and enter items using the items Barcode SKU lookup.

Press  to save the *Item Id* field.

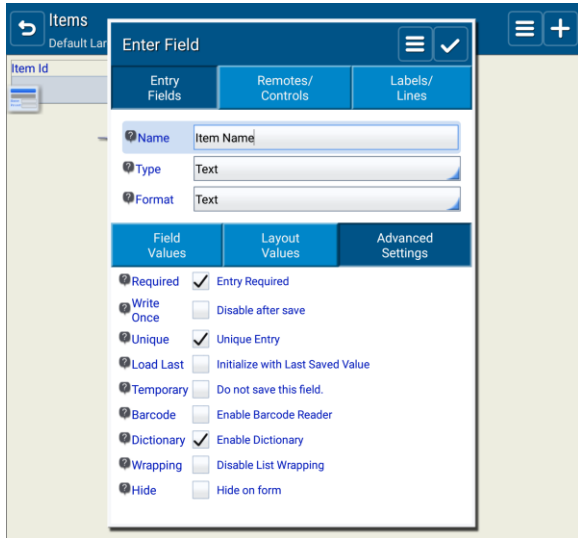
## Entry Field Advanced Settings


<b>Required</b>	Check to require that this field has a value and is not blank. You cannot save or replace this entry if this is blank. Enable for <b>Unique Keys</b> to make sure there is always a value.
<b>Write Once</b>	Use to disable this field after a value is saved. Use this to keep <b>Unique Keys</b> the same value. As an Unique Key, if this value changes, all of the remote connections to this entry would be lost.
<b>Unique</b>	Require that no two values in this field are the same. Use this for <b>Unique Keys</b> to guarantee that when searched, only one unique value is found.
<b>Load Last</b>	Use to initialize the field to the last saved value when starting a new entry. This is similar to using the <b>Init Math</b> and using the <b>last()</b> math command, except this occurs before the init math occurs.
<b>Temporary</b>	<p>Temporary allows you to create a field that is not saved, and it's value is expected to be calculated. Time sensitive values, such as Age calculation, will automatically be current when the entry loads.</p> <p>Read Only can only be set on a new field, since it does not create a Database Entry.</p>
<b>Barcode</b>	Enable the barcode reader to be used with this field.
<b>Dictionary</b>	If checked, the built in Android Dictionary will show when entering values. Handy when entering notes or comments.
<b>Wrapping</b>	You can disable this field from wrapping to multiple lines when show in the list. This can also be toggled in the list view. Use this for text fields that may contain a lot of text.
<b>Hide</b>	<p>Hide this field when viewing this form. Use for "scratch" values that you don't want shown. Menu Button commands can also <b>Show</b> and <b>Hide</b> fields.</p> <p>This is different than using the Hide  button for hiding fields on different form layouts.</p>



You can use the arrows  to move and resize the entry fields.

When entering items, you will be able to tap the barcode and bring up the barcode reader.



Press  to create the next field.

Enter *Item Name*.

Leave the Type as Text.


Tap the **Advanced Settings** tab.

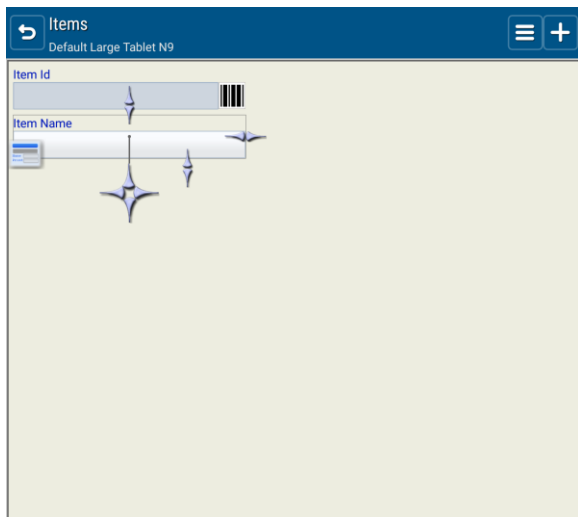
The *Item Name* will be used to easily search and lookup the item by name, if you don't know the Item Id.

Since this is not a Unique Key, but a value we want to search on, and would like to be able to change in the future if needed, check the check boxes, **Required** and **Unique**. These will force this to always have a Unique Value, but let you change it in the future.

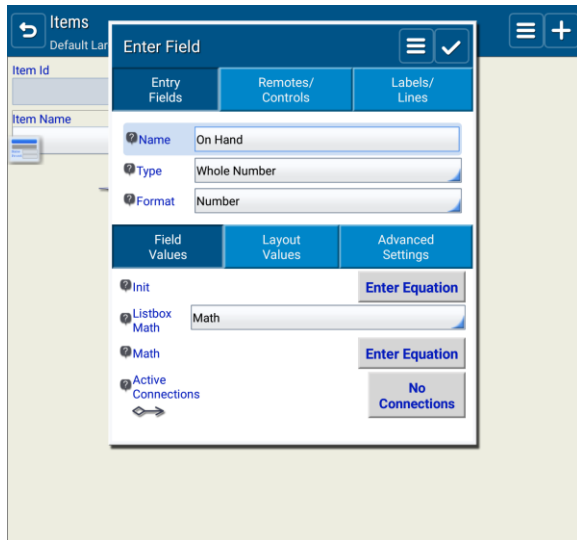
If you do not set **Unique**, it will still work, but it could be confusing when looking up multiple items with the same name.

Enable Dictionary will help you look up words when entering the name.

Press  to save the *Item Name* field.



Now let's add some number fields to keep inventory quantity.

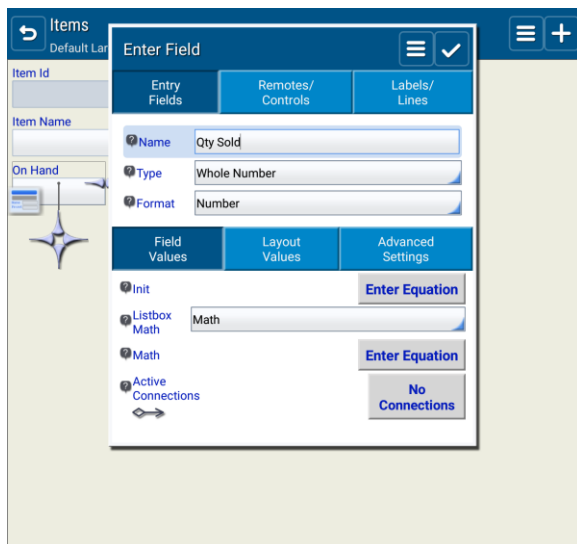
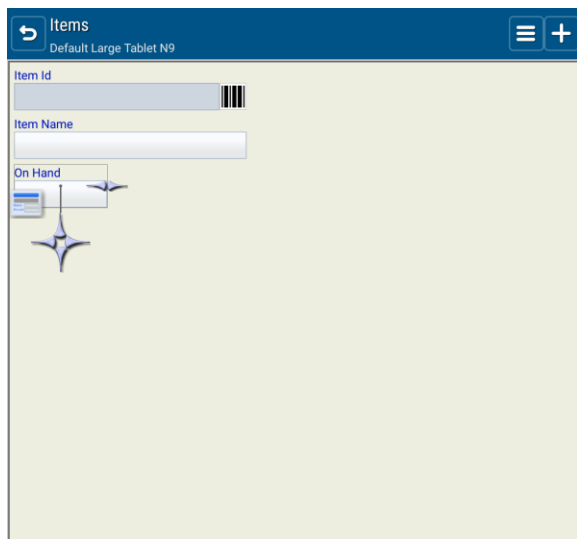


These will be used to keep track of our Inventory Levels and Sales history.

Add a Quantity *On Hand* number field.

Set the type as a number, we picked Whole Number since these are whole quantities. You could use Decimal numbers if you are selling fractional or decimal quantities.

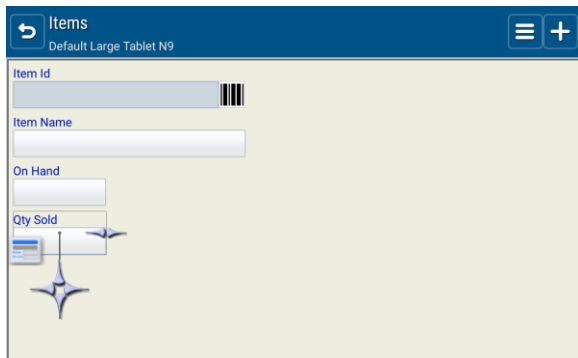
Save the *On Hand* Field.



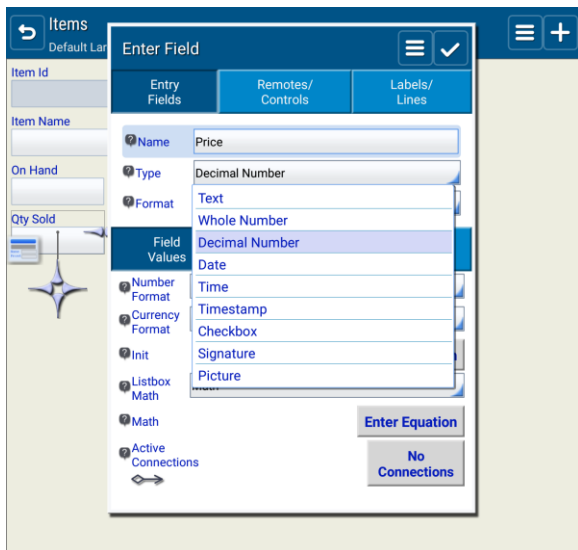
Now let's add a *Qty Sold* field.

Setup *Qty Sold*, the same as Quantity on Hand.

Save the *Qty Sold*.

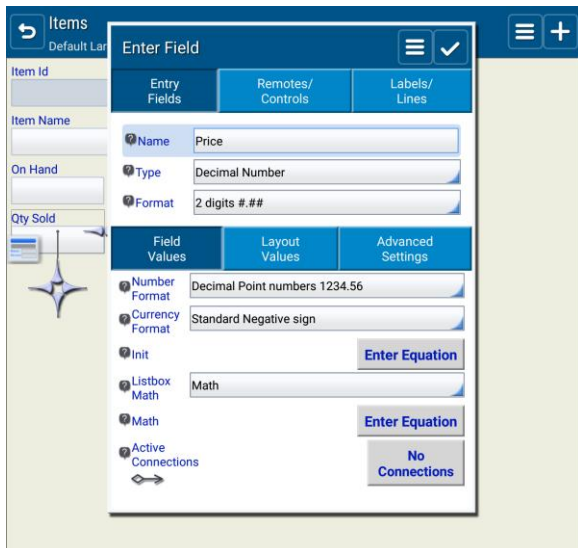


Finally, add a *Price* field so we know what we charge for this item.



Typically this is a Decimal Number, 2 digits to store currency values.

You should check the **Required** checkbox to force that there is always a price entered.



### Number Format

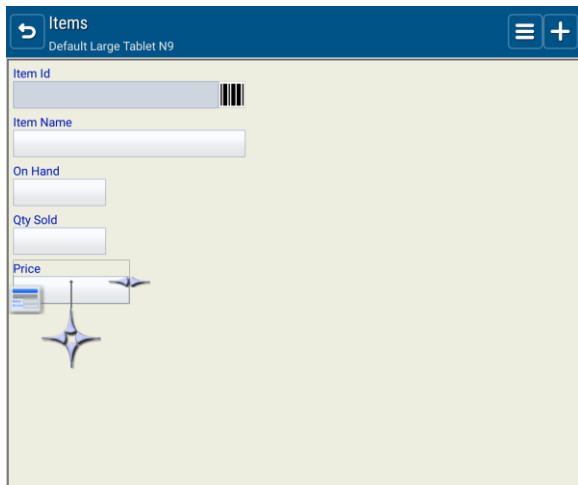
You can pick the format of commas and decimal points in Decimal Numbers.

### Currency Format

You can define how negative numbers appear and where an optional Currency symbol appears.

### Currency Symbol


Enter any optional symbol you would like to add to the number. Use Currency Format to set if it is added at the start or the end of number.

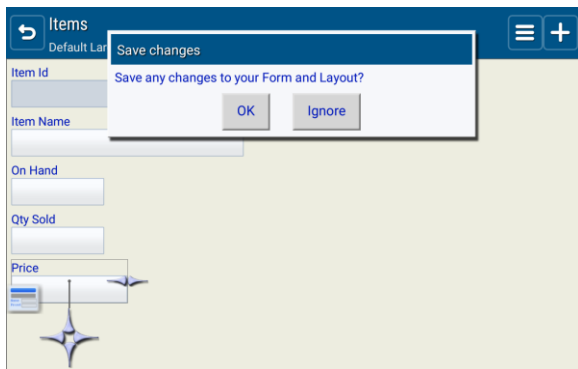


Save the Price field.

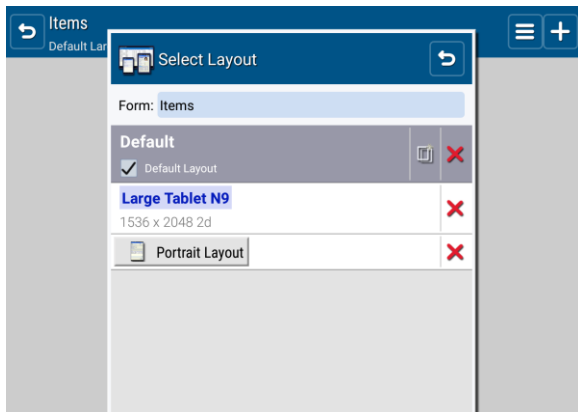
Our basic Inventory Item Form is complete.


You would add more fields, like cost, quantity on order, Total Cost, and Total Price, Taxable, Weight, a Picture, but for now we will keep it simple.

Press  and save the changes for the Items form.



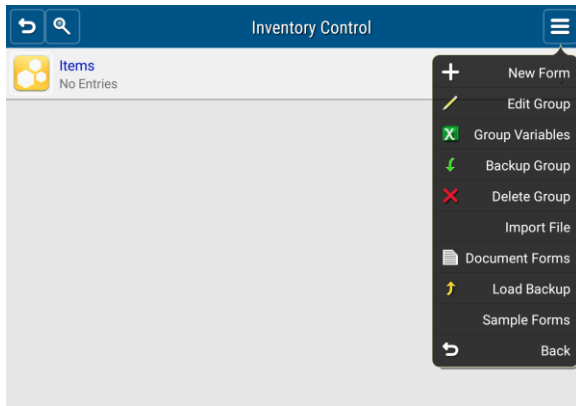
Press OK to save the changes.



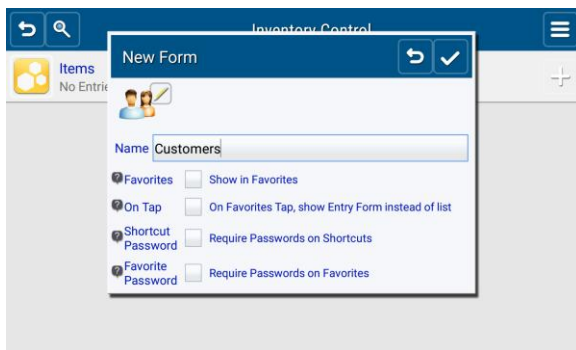
Press  to exit the form layout selector.

Now let's create a customer form.


## Customer Form

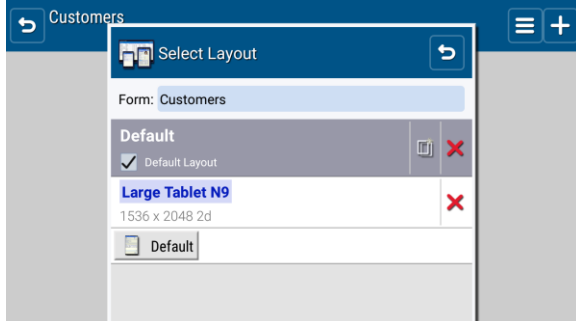


Press menu and select **New Form** to create a new Customers Form.

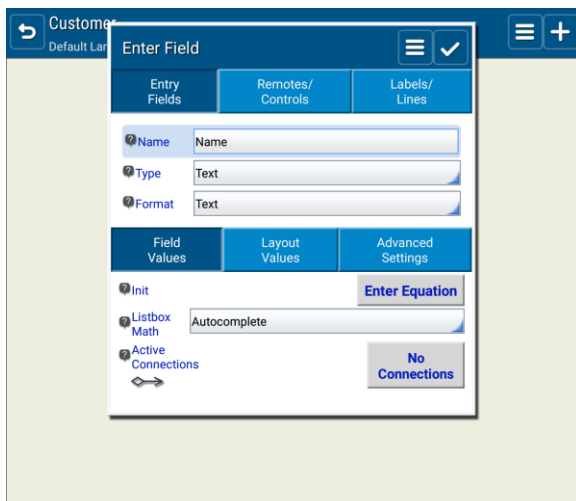



Enter *Customers* for the name and pick an Icon if you wish.

Press  to save the *Customers* form.



Select the **Default** layout.



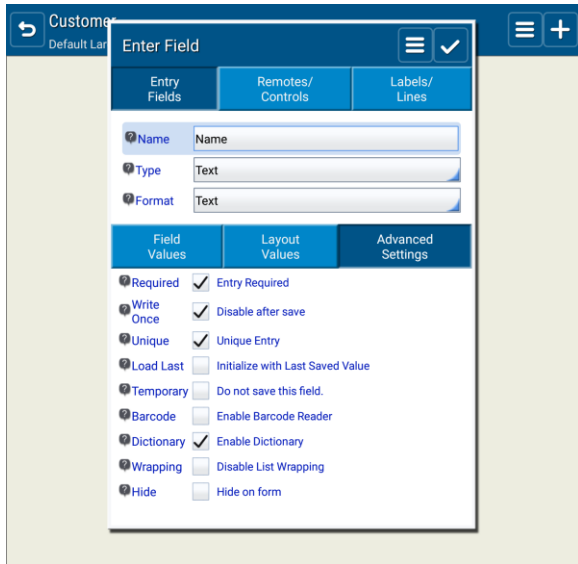
On the blank new customer form, press  to add a new field.

Add a Text Field for the Customer's Name.

Enter *Name* for customer field name.

Press the **Advanced Settings** Tab.



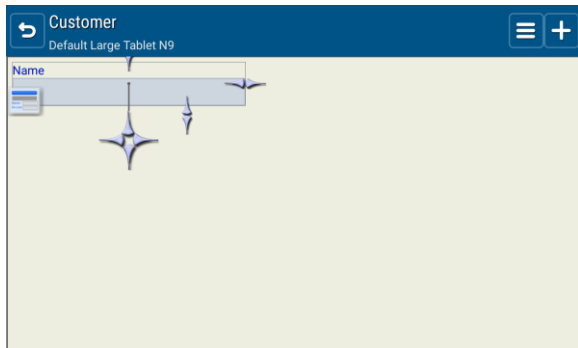


Make the Name field a **Unique Key** by setting the checkboxes **Required**, **Write Once**, and **Unique**.

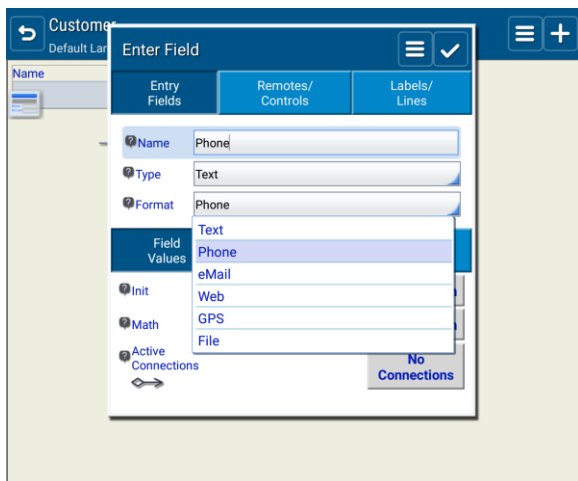
This will force this field to always have the same value, that never changes (**Write Once**), that has no matches (**Unique**) and has a value (**Required**).

Dictionary controls whether the word suggestions show when the keyboard is shown.

Save the *Name* field.



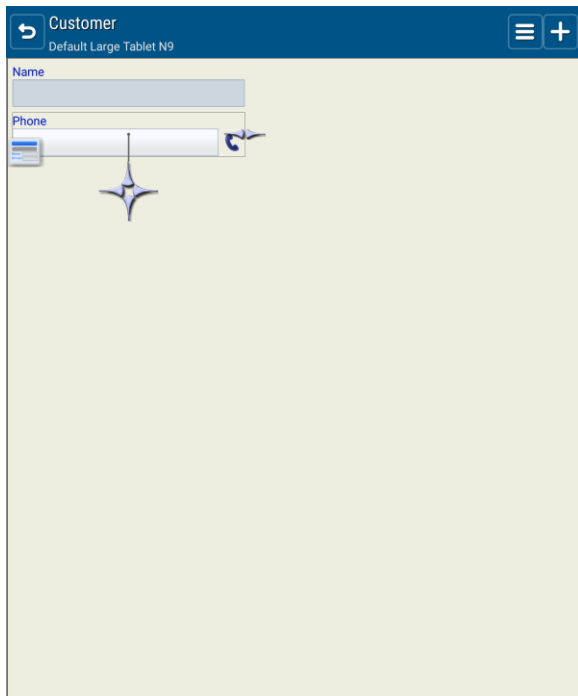
Now let's add another field, Phone number Text field.



Set the **Format** to Phone.

We will be able to see this field in the Invoice and look up the customer by Phone Number or jump to phone dialer to call the number.

Save the *Phone* field.



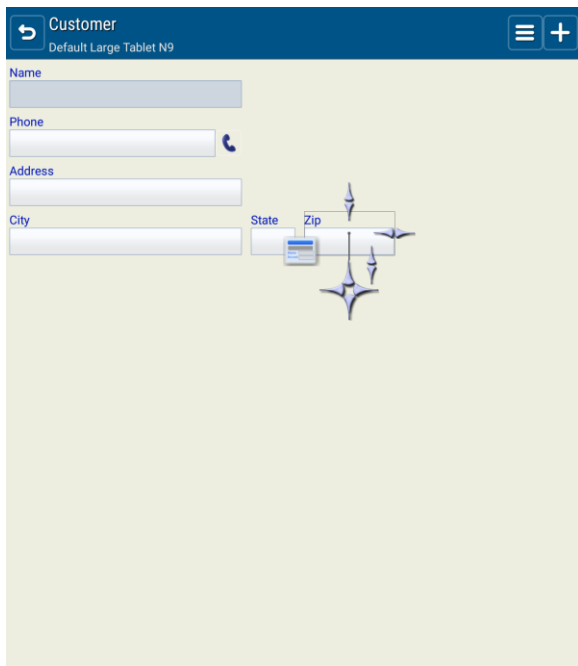
The Customer form with Name and Phone number

When viewing and entering entries, you can tap the Phone icon to bring up the phone dialer.


Add four more text fields to save the Address.

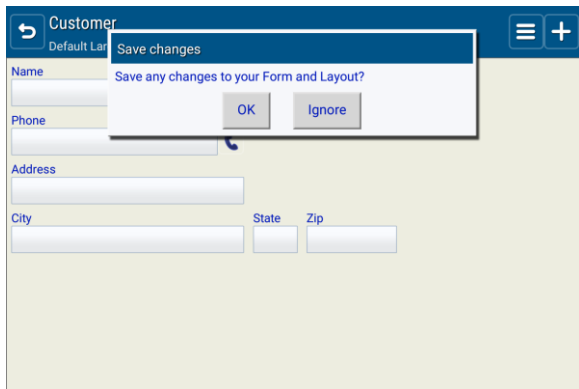
- Address
- City
- State
- Zip

We will connect to these fields so they can be shown and used in the Invoice.

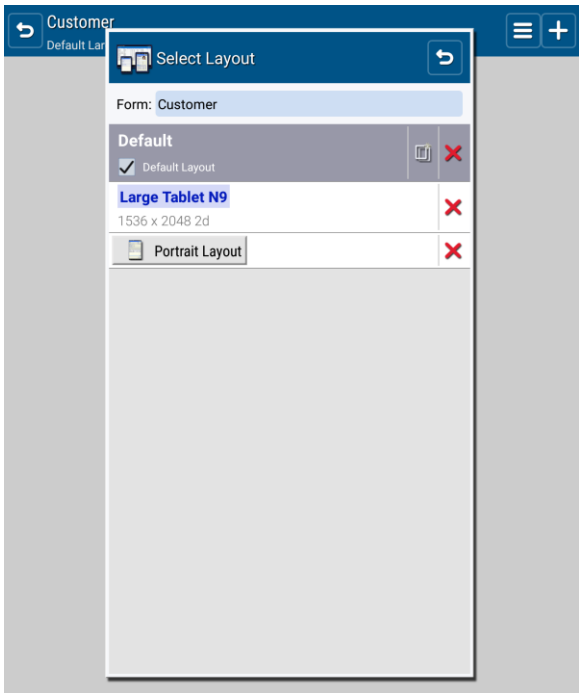



Use the field dragger  to move the State and Zip fields to the same line as City.

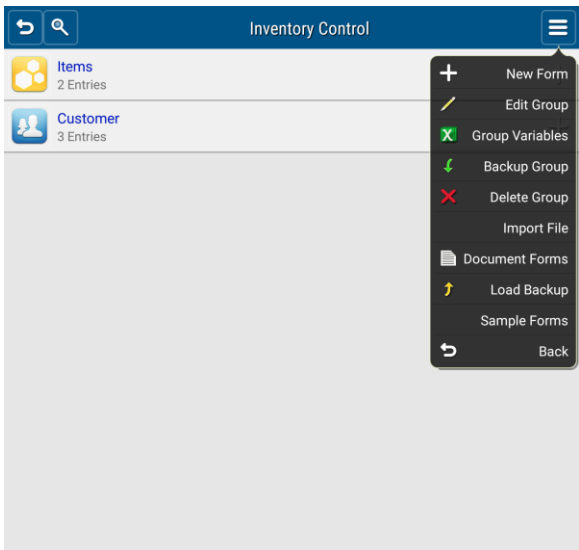
Press  and save the changes for the Items form.



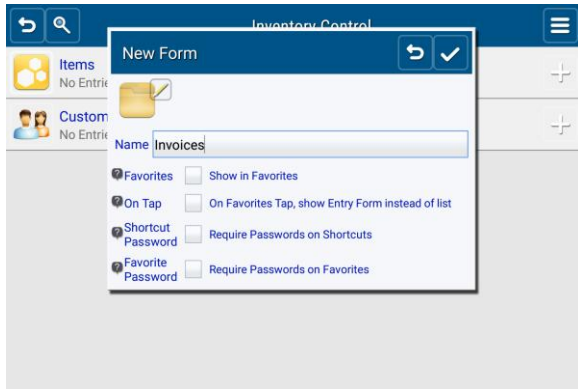
Press OK to save the changes.




Press  to exit the form layout selector.



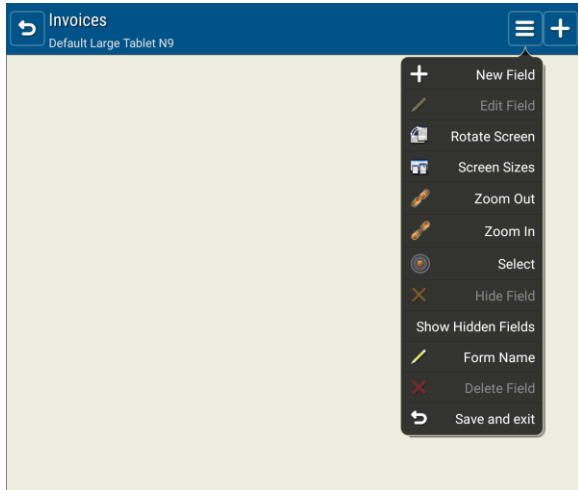
Press menu and select **New Form** to create a new Invoices Form.




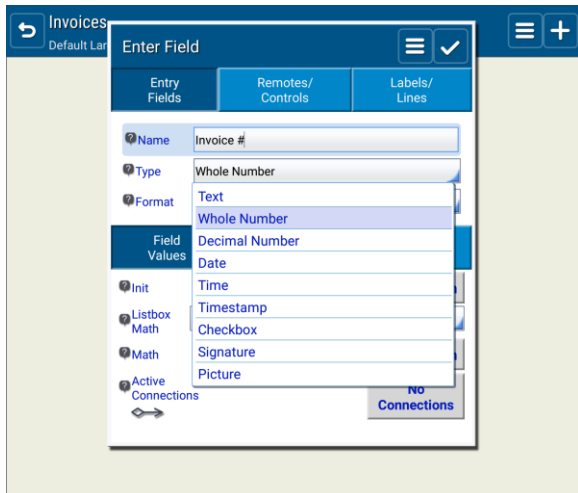
Enter a Form Name, *Invoices*.

Tap the Icon  to pick an Icon for this form.

Press  to save the new Form.



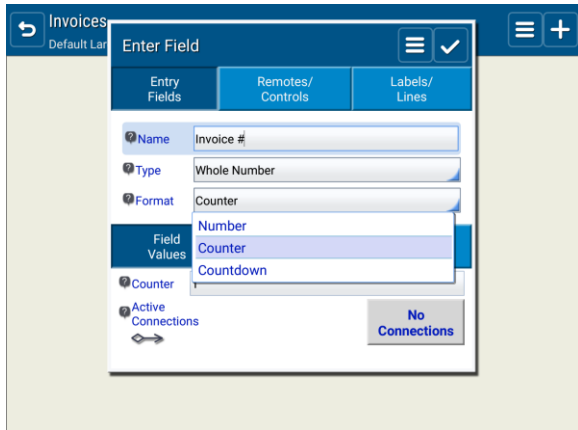
Press  to create a new Field.



Enter *Invoice #* for the name.

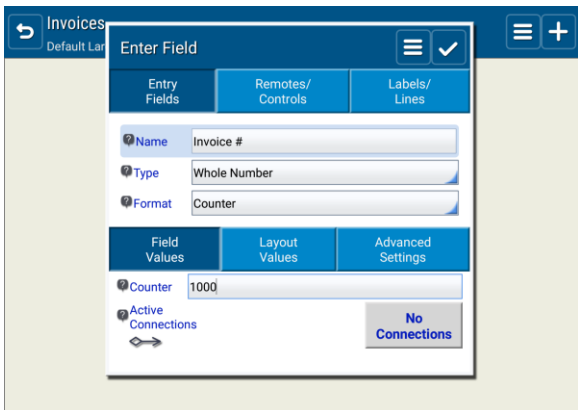
Select the **Type** Whole Number.


Whole numbers do not have decimal values, and can also be an Automatic Up or Down counter.




Select the **Format** Counter.

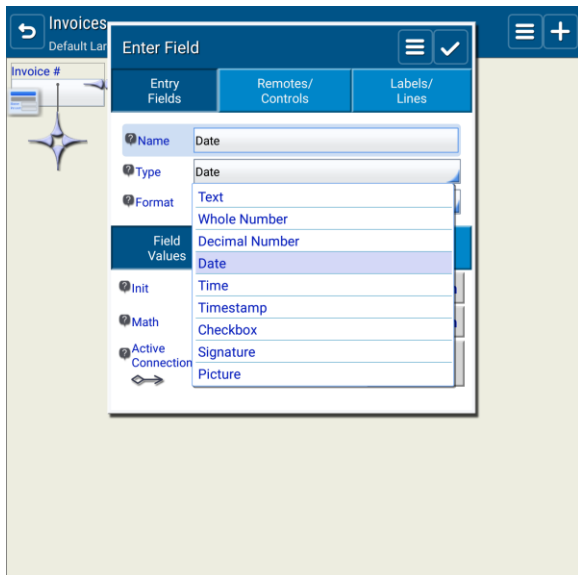
Change the **Counter** value from 1 to 1000.



Press  to save the new Invoice # Field.



Press  to create a new Field.



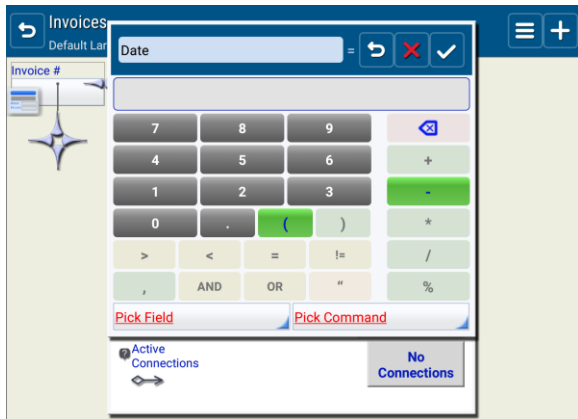
Enter *Date* for the name.

Select the **Type** Date.

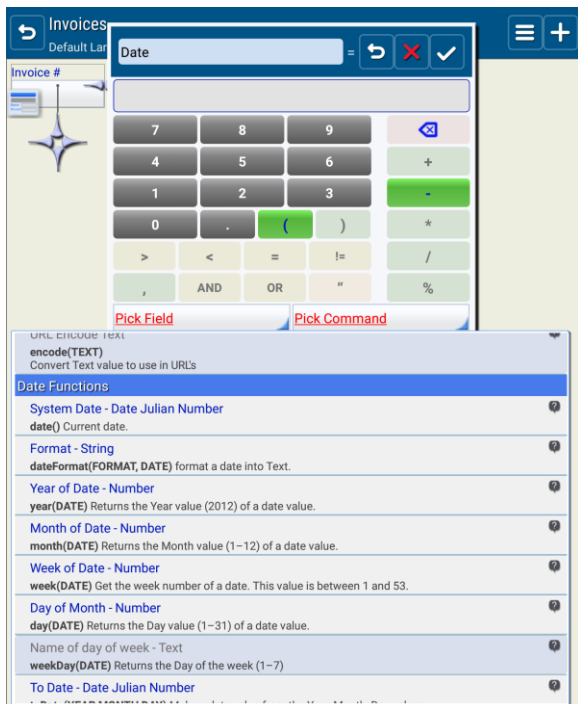
Keep the **Format** Local Format.

The local format will use the default date format that the device uses. You can change the format at any time.

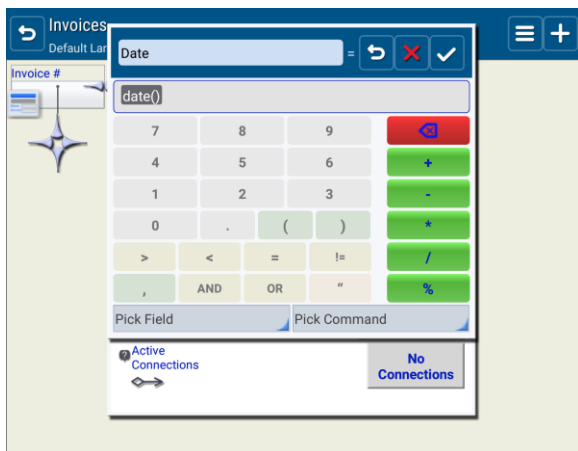
We want to initialize the date to the system date, so press **Enter Equation** for the **Init** math.




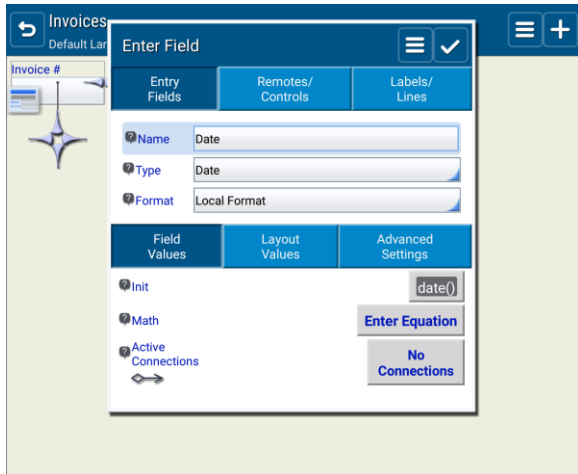
Press **Pick Command** on the math page.




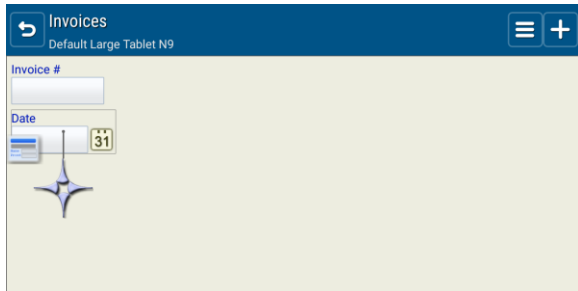
Scroll down to the Date section and select **System Date**.




Press  to save the init math equation.

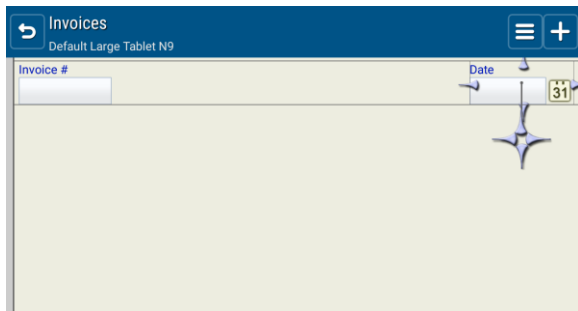



Press  to save the new *Date* Field.

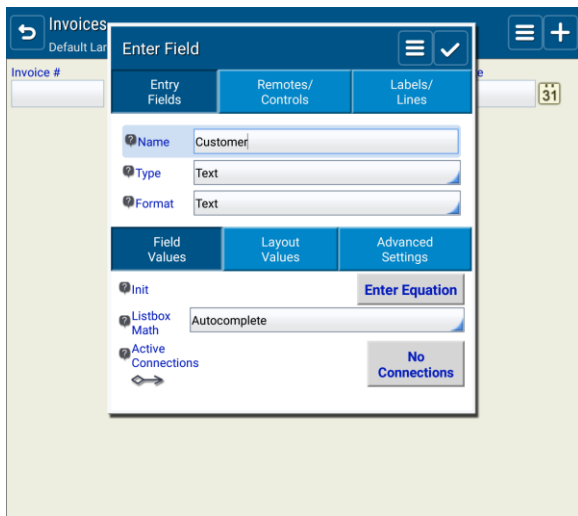


Use the field dragger  to drag the new Date Field to top right of the screen.

When entering the date, you can press the Date Icon to bring up the calendar.



Press  to add a new entry field.



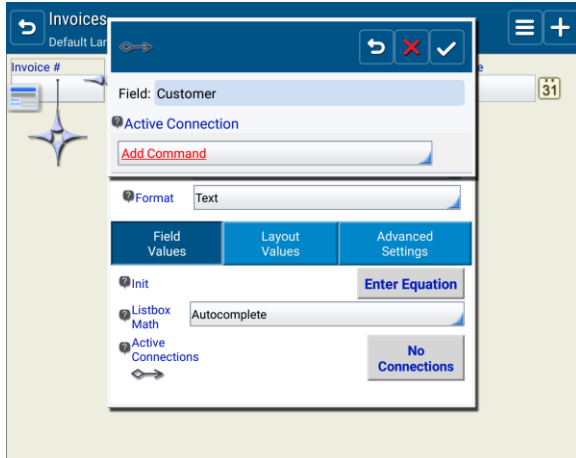
Leave the type as Text and set the name to *Customer*.

We are going to connect the Invoice to the Customers form.

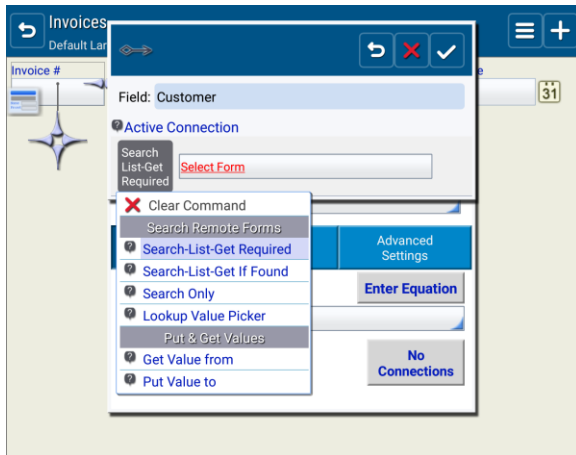
Click the **Active Connection** button that says **No Connections**.

## Active Connections

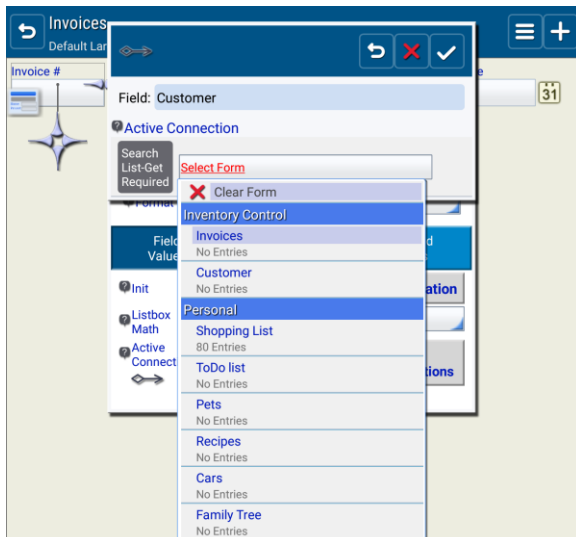
To connect the Invoice form to the Customer form, we need to define an **Active Connection**.



Select **Add Command**.



Select **Search-List-Get Required** in the Search Remote Forms commands.

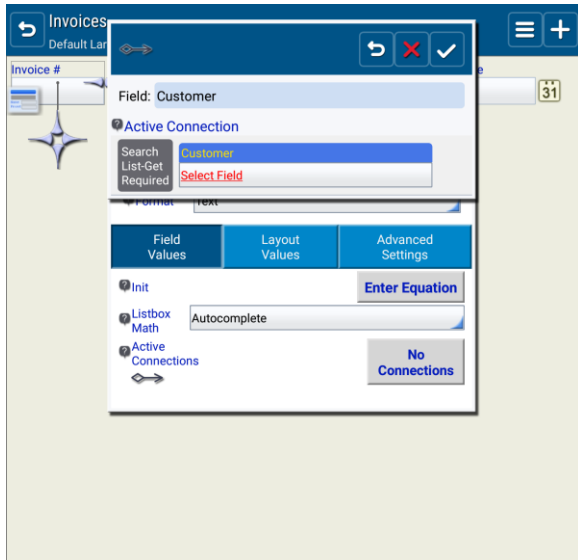


Tap **Select Form** and pick the **Customer** Form.

**PRO TIP:** All of the Forms in your Group of Forms will be shown first to make it easy to pick your forms.

Pick **Customer**.

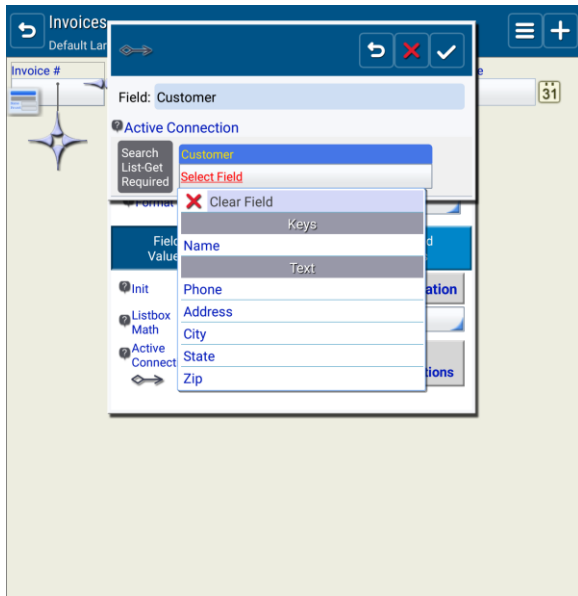




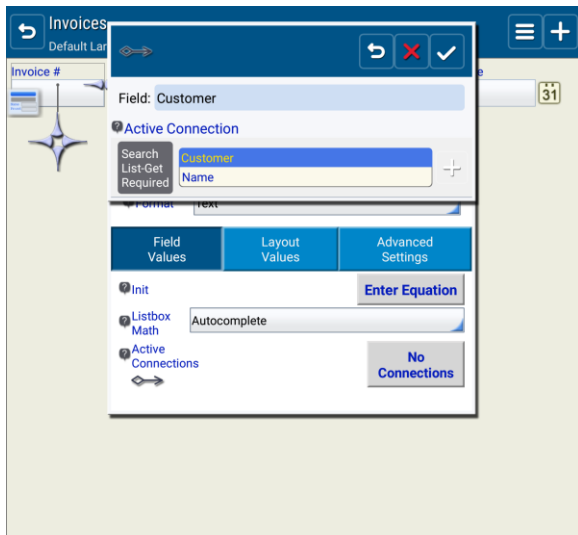
Next, select the field you want to search.

Press on **Select Field**.

You should always try to pick **Unique Keys**, since their value does not change and the link between forms will be stable and the Invoice will be able to always find this customer entry for reporting.

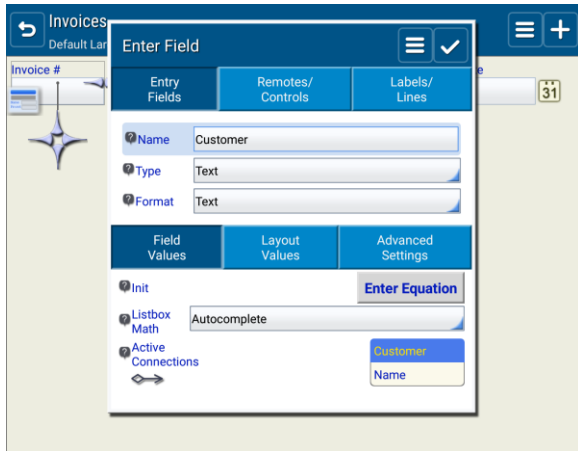



Pick **Name** since it is the **Unique Key** for our customers.

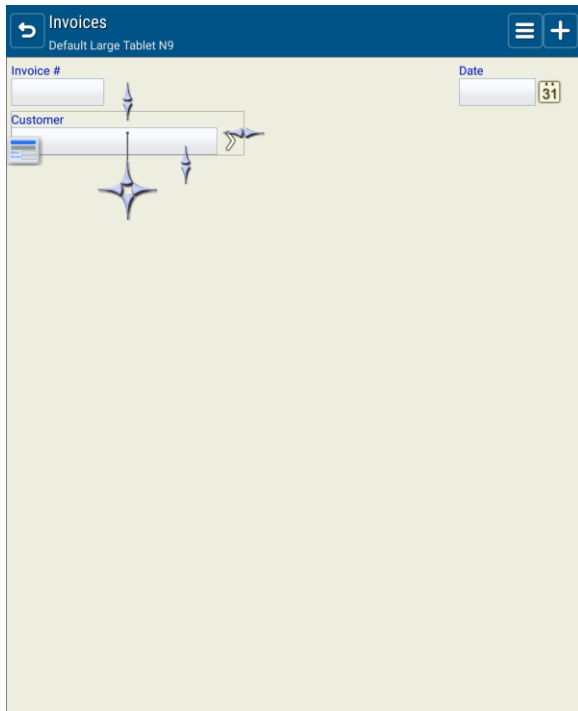


You have now created an Active Connection between the two forms.

Press  to save the connection.



Press  to save the field.




Our Invoice form with the customer Search-List-Get **Active Connection**.

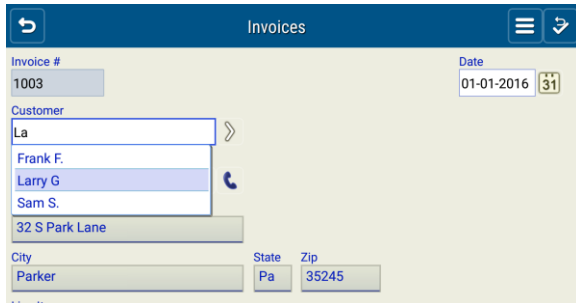
## Search-List-Get Required and Search-List-Get If Found

**Search-List-Get Required** is used to force that the value entered connects to a valid remote Form. If left empty, it is still valid. Use the Required checkbox if you want to make sure a value is entered.


**Search-List-Get If Found** is used when you do not require a valid value. This allows "one off" type entries where it does not matter if a remote entry is found. Even if there is no existing link, you can still save the entry.


A field with **Search-List-Get Required** is used to lookup and keep the connection between the two forms. This connection is used when reporting and doing other field Active Connections such as **Get from's** and **Put to's**.

Remote Forms that are connected will have  in front of the form name when picking forms.



The screenshot shows a mobile application interface for 'Invoices'. At the top, there's a blue header with a back arrow, the title 'Invoices', and a menu icon. Below the header, there are several input fields: 'Invoice #' with the value '1003', 'Date' with '01-01-2016' and a calendar icon, and 'Customer' with a dropdown menu. The dropdown menu is open, showing a list of names: 'La', 'Frank F.', 'Larry G.', and 'Sam S.'. Below the dropdown, there's a text field with '32 S Park Lane'. At the bottom, there are fields for 'City' (Parker), 'State' (Pa), and 'Zip' (35245). A 'Spy Glass' icon is visible at the top right of the form area.

You can jump to the selected form when entering values. Press  at the end of the field to jump to the remote form.

If this entry is empty it will let you search in the remote form by pressing the Spy Glass  at the top of the screen.

Once a remote entry is selected, press  to select the entry and return back.

If there has been a entry saved, you can only go to that entry.

If there has been a entry saved, but this field is not **Write Once**, you can lookup and select another entry.

If this is **Write Once** you cannot lookup once the value has been saved. **Write Once** fields should be used with Connections that you do not want changed in the future.



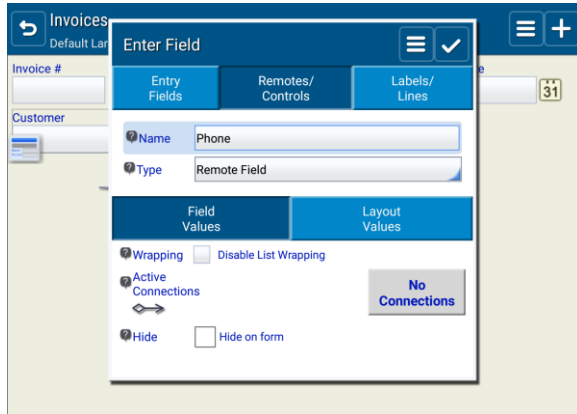
## Active Connections for saved Fields

<b>Search List Get Required</b>	Typing in the field will find the closest matches in the remote form. A valid value must be selected to save the entry.
<b>Search List Get If Found</b>	Typing in the field will find the closest matches in the remote form. Non matching values can be saved, but no connection link will be established.
<b>Search Only</b>	Use to do secondary searches to other forms using the same key. This allows the same key to be used with multiple forms. See the <i>Multiple Keys Pro Sample</i> .
<b>Lookup Value Picker</b>	Use to Lookup Values from remote Forms, without linking to the remote record. This let's you find and enter a remote value that can be used for key lookup or button commands to transfer to fields, without a direct connection to the remote form. See the <i>Dual Transactions Pro Sample</i> .
<b>Get Value From</b>	If the remote form has a Search-List-Get connection, you can get other field values and save them in this field. If the remote value changes, it will try to change the value in this field. If you don't want that to happen, Check the <u>Write Once - Disable after save</u> checkbox to block this field from getting the remote value.
<b>Put Value to</b>	If the remote form has a Search-List-Get connection, you can put and save the value in this field into the remote field.
<b>Add To</b>	<p>If this field is a number, you can add the value of this field, to a remote number field. When the entry is saved, the value in this field is added to the remote field. If this value is changed later, just the difference is changed in the remote field.</p> <p><b>Example:</b> Say this field is the number of items sold and set to 5. When saved, 5 is added to the remote field value. If the value is change to 6, then 1 is added to the remote field, or if the value was changed to 4, then 1 is subtracted.</p> <p><b>Note:</b> If this record is deleted, then all of this value is subtracted from the remote value. You can use <b>Delete without Updates</b> to delete an entry without this automatic update.</p>
<b>Subtract From</b>	<p>If this field is a number, you can subtract the value of this field, from a remote number field. When the entry is saved, the value in this field is subtracted from the remote field. If this value is changed later, just the difference is changed in the remote field.</p> <p><b>Example:</b> Say this field is the number of items sold and set to 5. When saved, 5 is subtracted from the remote field value. If the value is change to 6, then 1 more is subtracted from the remote field, or if the value was changed to 4, then 1 is added.</p> <p><b>Note:</b> If this record is deleted, then all of this value is added to the remote value. You can use <b>Delete without Updates</b> to delete an entry without this automatic update.</p>

## Embedding Remote Fields

Embedded Remote Fields let you have a remote field value in this form, but it is never saved in this form, only looked up from the remote form.

This is ideal for values that you want to see, like an address, phone number, or note that acts like it is part of this form. It can be used in math equations, shows in lists, can be sorted, filtered and reported, but the value is always the value from the remote Form's Field.

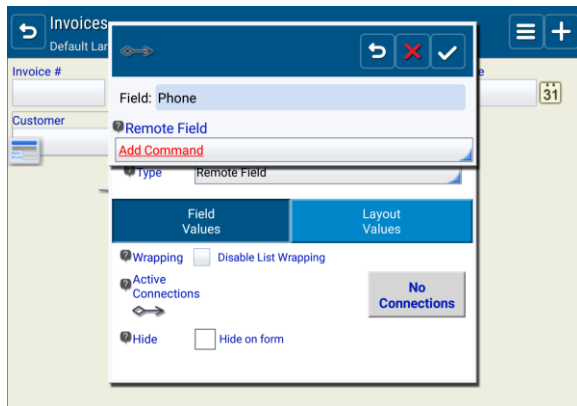


To add the *Customer Phone* to the Invoice we will use a **Remote Field**.

Add a new field and select the tab **Remotes/Controls**.

Enter the field name, *Phone*.

In **Remotes/Controls** there are seven field types and a button.

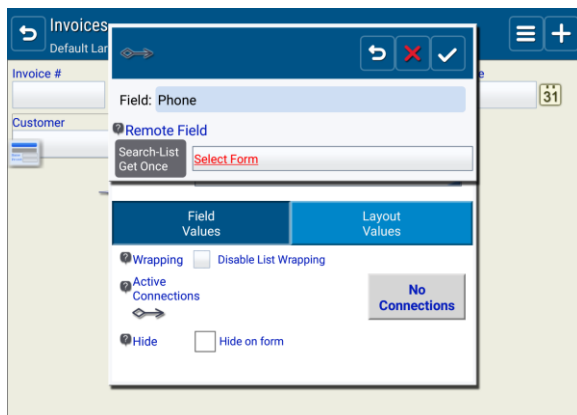


Pick **Remote Field**.

Click the Active Connections button that says **No Connections**.

Tap **Add Command**.

Pick **Search-List-Get No Lookup after save**.



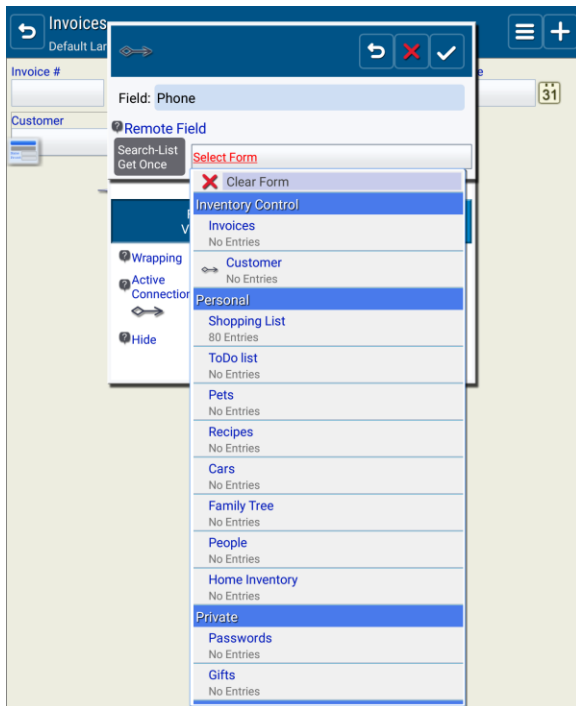
## Search-List-Get No Lookup after save

The Remote field has two Search-List-Get functions.

The first one, let's you always Search-List-Get before and after the entry is saved. Use this when you don't care if the remote link changes to another entry. When you edit a saved entry, you can use this field to lookup another value, which may change which remote entry that this entry is linked to. To avoid that, use the Search-List-Get No Lookup after save connection link.


The Second one, Search-List-Get No Lookup after save is used when you don't want the remote link to be changed after the entry is saved. This is like putting the Write Once flag on your Customer Name field.

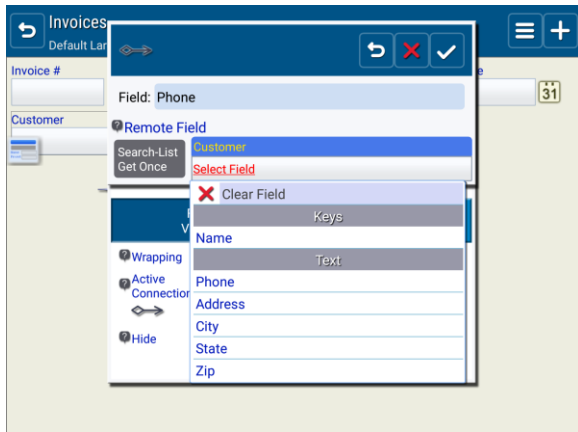
Select **Search-List-Get No Lookup after save**.



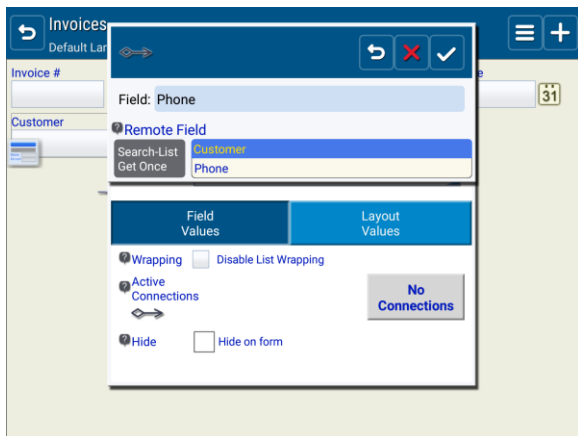
Tap **Select Form**.

Now we will select the **Customer** Form to search.

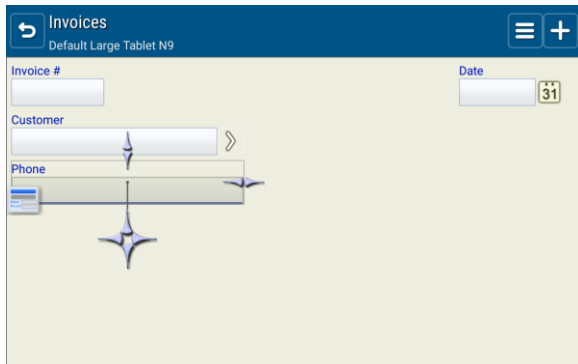
Notice the  in front of the Customer form. That shows we have a connection to that form.



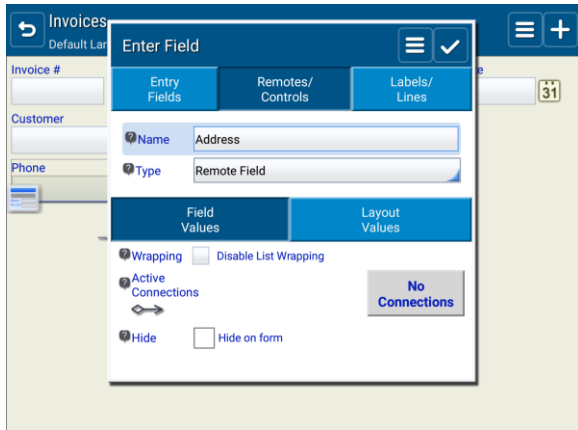
Pick the **Phone** field from the Customer Form to **Search-List-Get Once**.



Save the Active Connection and save the new Remote Field *Phone*.



The new Remote Field *Phone* on your Invoices form.



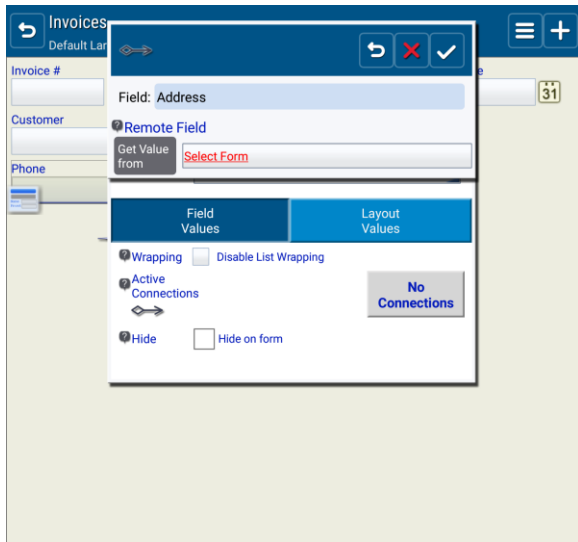
Now let's hook up the Address Field.

Add a new field and name it *Address*.

Tap the **Remotes/Controls** tab.

To get the Address from the Customer to put on the Invoice we will use a **Remote Field**.

Press the Active Connections button **No Connections**.

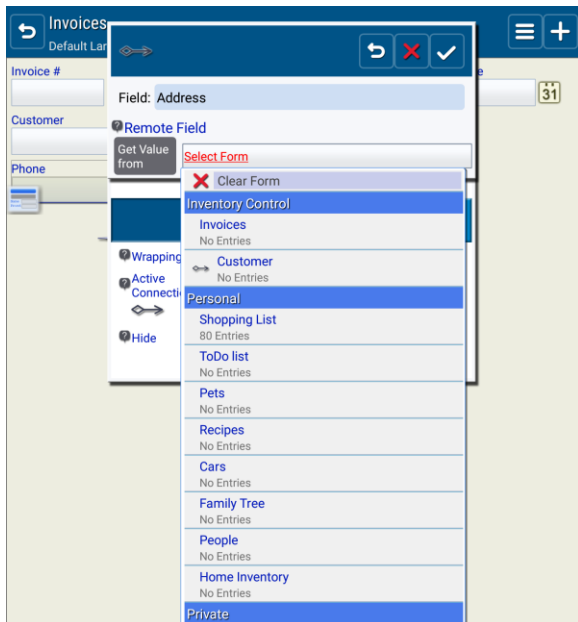


Pick **Get Value from** to get the address from the customer form.

This value will always display and show the value that is contained in the customer entry address.

If the address changes, this will always show the current value.

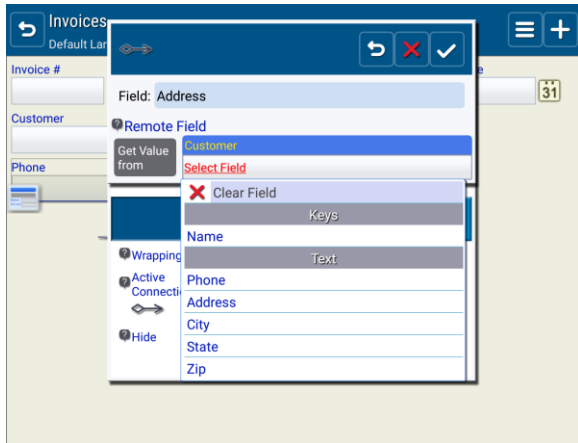
Tap **Select Form**.



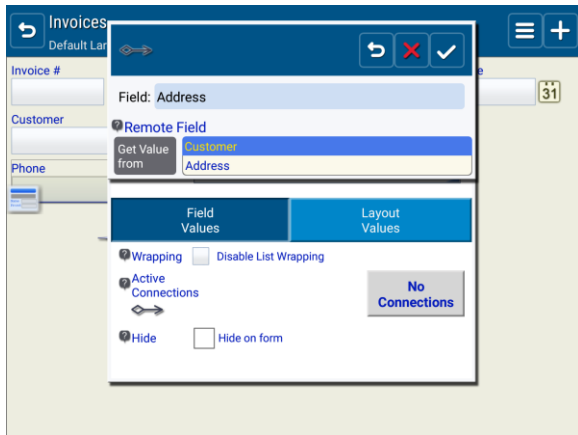
This value is not saved in the Invoice entry, it is always looked up, using the Customer's Name **Active Connection**.

Select the **Customer** Form.

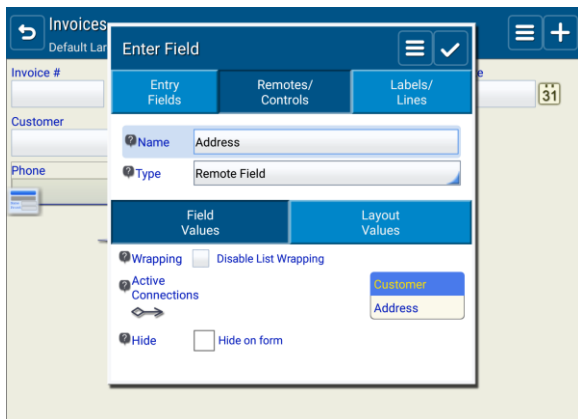




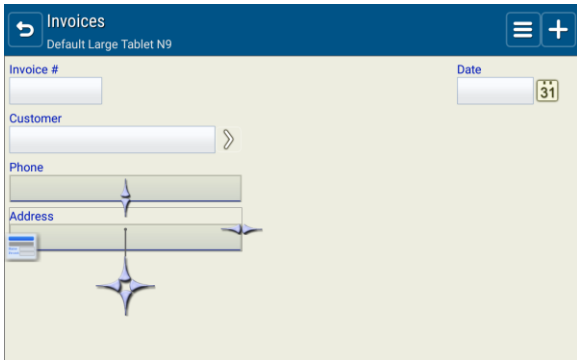
Select the **Address** field.



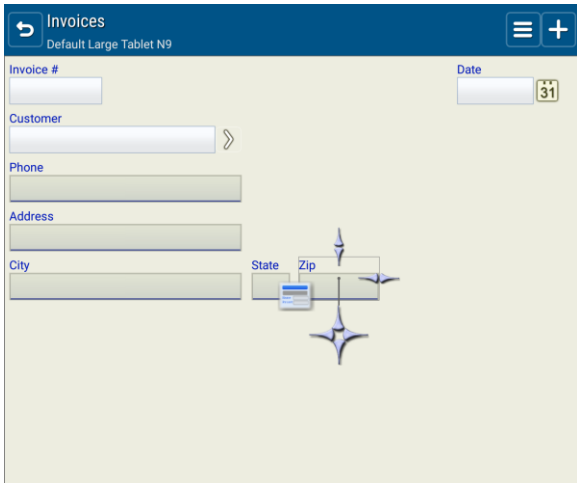
Save the connection.



Save the new Remote *Address* Field.




Repeat the same steps for the *Address* to get the *City*, *State* and *Zip* remote fields.

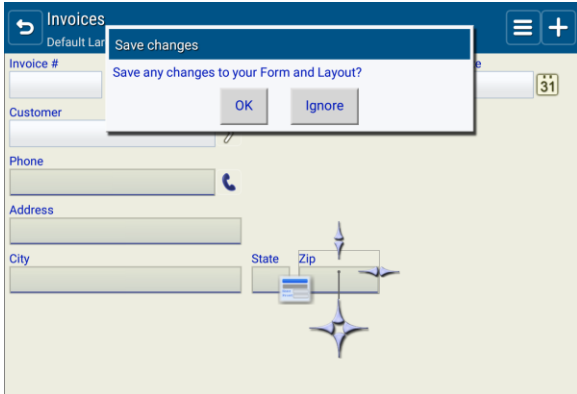


Our Invoice is now connected to a customer.


You can lookup by the **Unique Key** Customer Name, or by using their Phone Number as a secondary lookup.

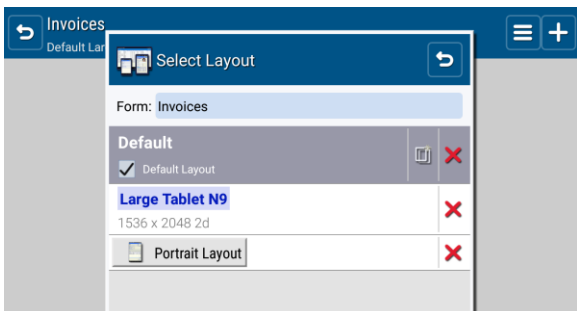
Their Address will automatically be filled in from their address fields.

Press  and save the changes for the Invoices form.



Press OK to save the changes.

Press  to exit the form layout selector.

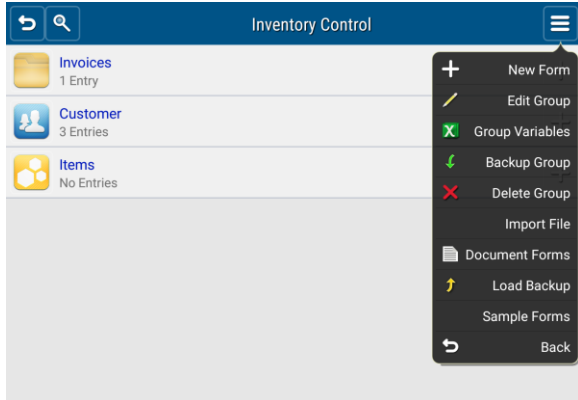


## Invoice Line items, Items Sold Form

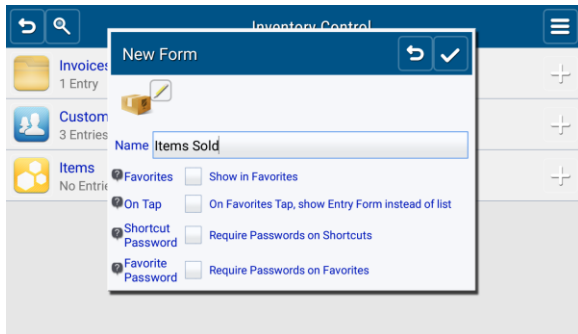
To connect an Inventory Item to an Invoice, we need an **Invoice Line Item**.

This represents a transaction. Quantity of an Item sold on a specific Invoice.

One entry gets created for each line item.



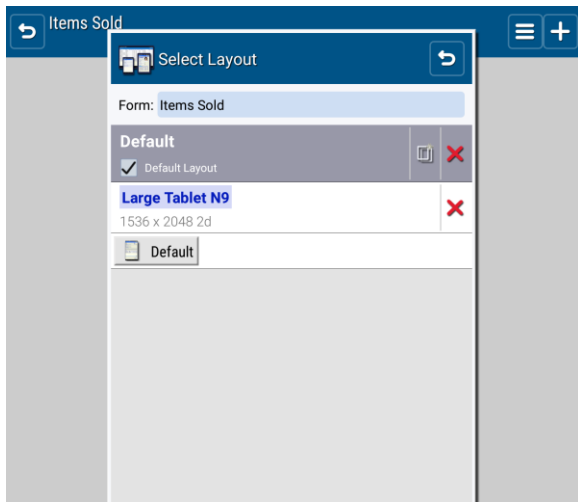
Create the Form, *Items Sold*.



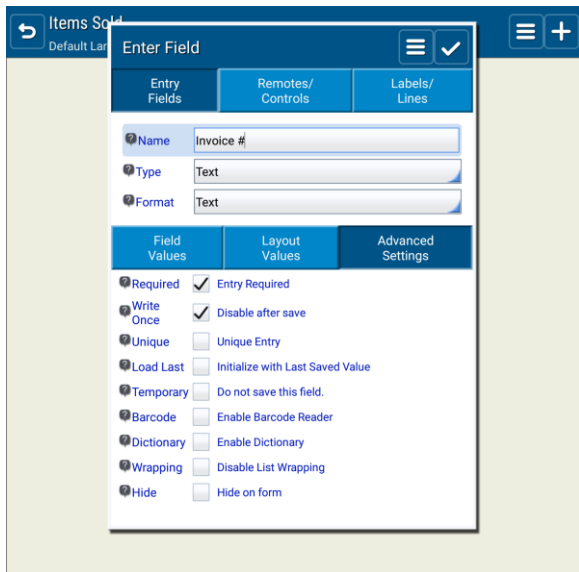
Name it *Items Sold*.

Pick an Icon

Save it.



Pick the **Default** layout.



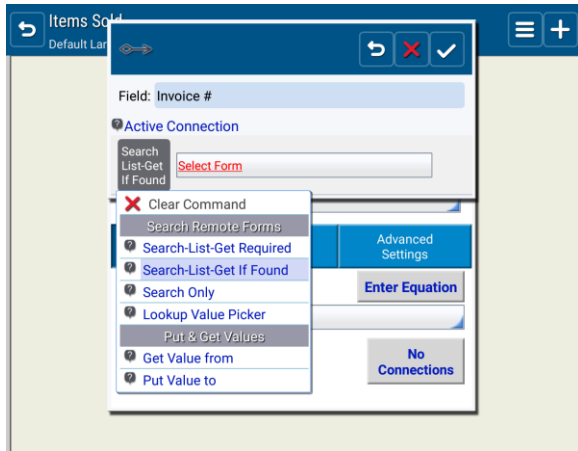
Add an *Invoice #* field. This is the Key value that the invoice uses to find all of it's entires.

You can have this search back to the Invoice as a convenience when viewing this entry.

Use Advanced Settings to set the checkboxes, **Required** and **Write Once**.

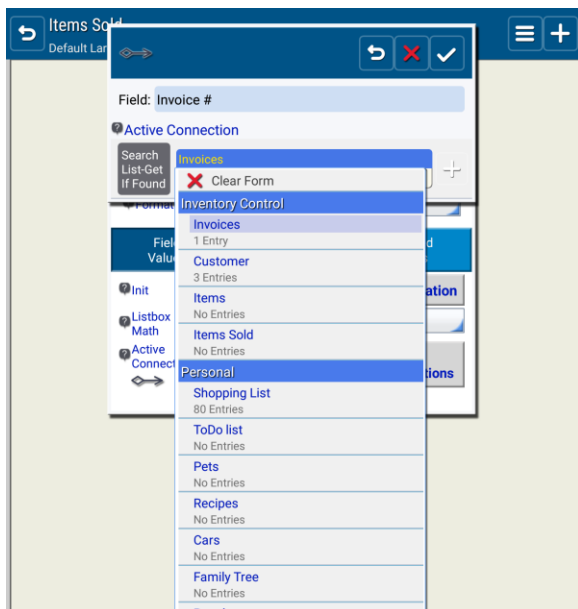
This will force an Invoice # to be saved and not changed later. The Invoice uses this Common Key Value

to find all of the items sold on one invoice.



Tap on **Field Values** and press the **No Connections** button.

By adding a **Search-List-Get If Found** you link the item sold to the Invoice and you can jump to the Invoice that this item transaction was sold on, as a convenience.



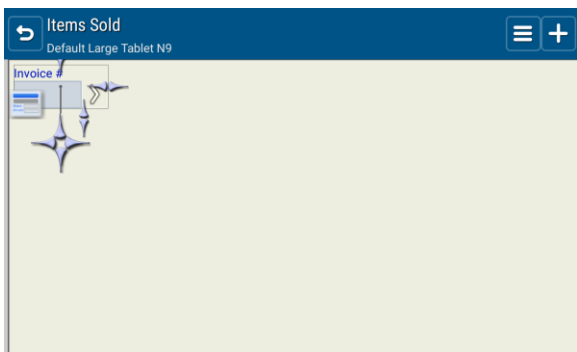
Pick the **Invoices** form.



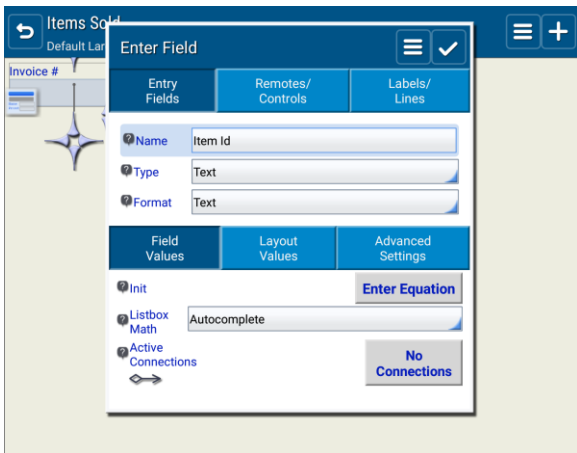
Pick the **Invoice #**.

Save the Active Connection.

Save the *Invoice #* field.

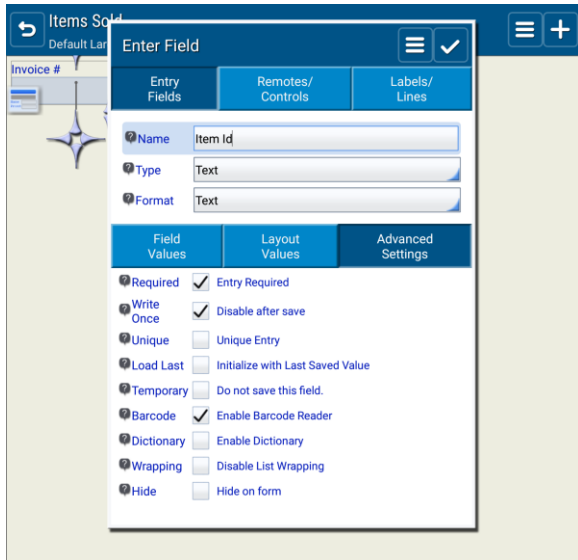


Next, let's create a new *Item id* SKU lookup field.



Add a new field and Name it *Item Id*

Click the **Advanced Settings** tab.



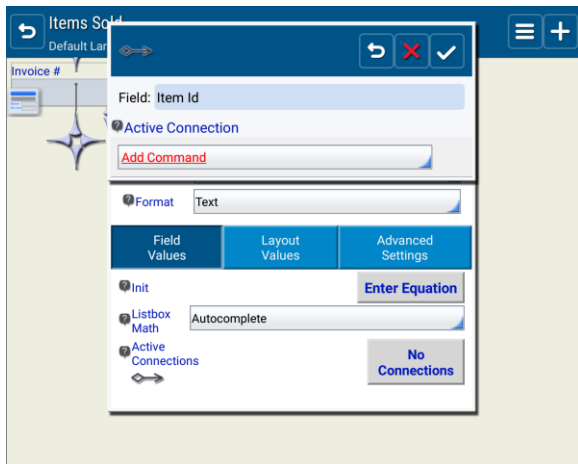
Check the **Required** and **Write Once** checkboxes.

**Required** will force a value which is necessary to create the Connection.

**Write Once** will keep the value from changing. If you make a mistake during Order Entry, just delete this entry.

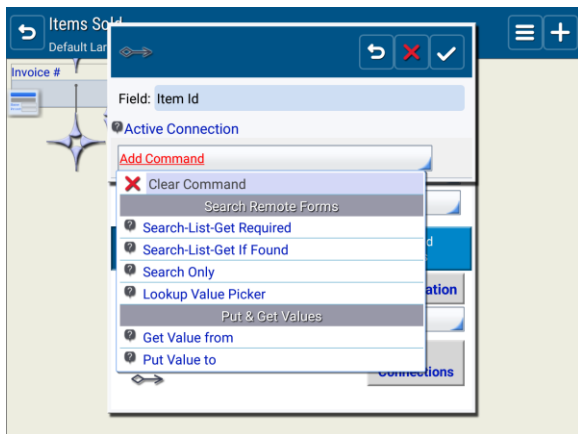
The **Active Connections** will automatically adjust the inventory quantities. If you allow users to change the key (Not checking **Write Once**), you lose that ability.

If you want to be able to lookup this item with the barcode reader, check the **Barcode** checkbox.

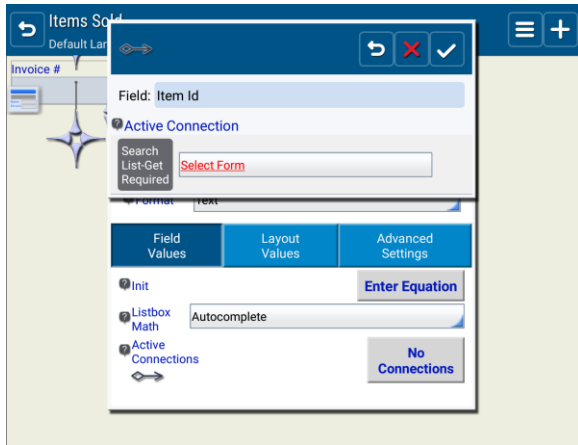


Click the **Field Values** tab and press the **No Commands** button.

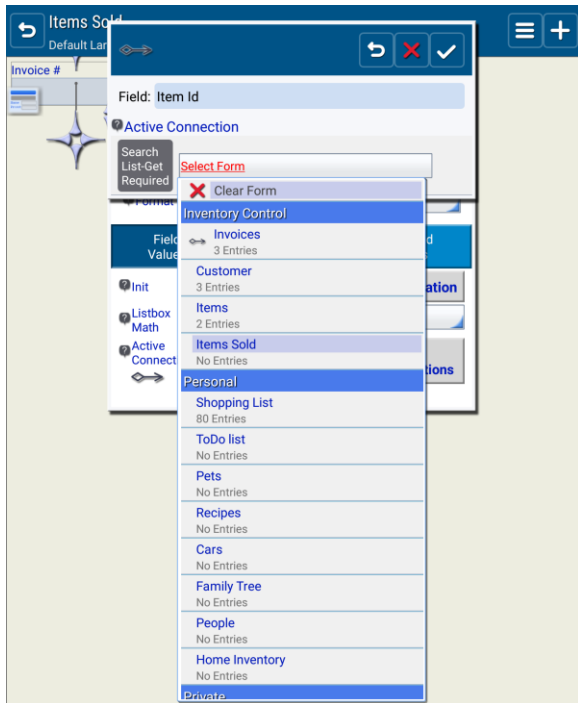
Press **Add Command**.



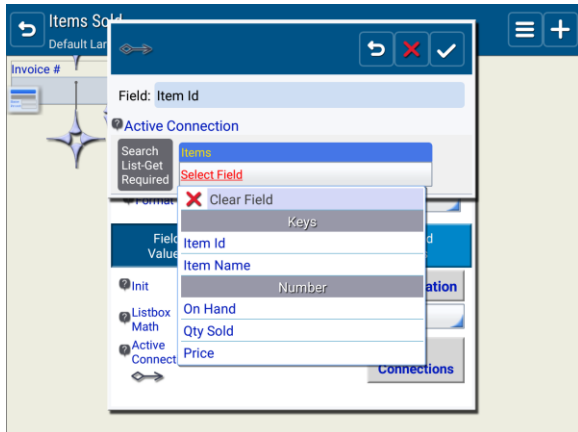
Select **Search-List-Get Required**.



Tap **Select Form**.

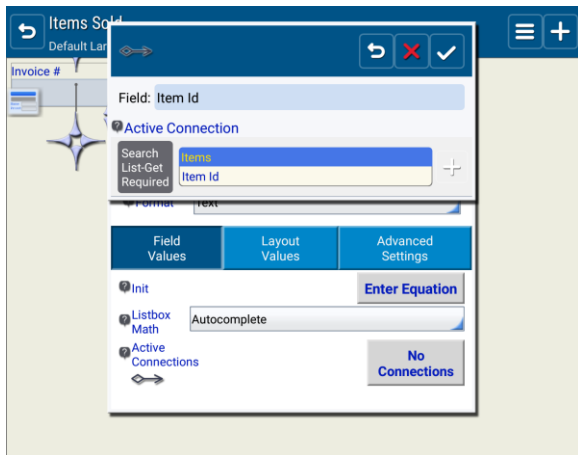


Pick the **Items** form.

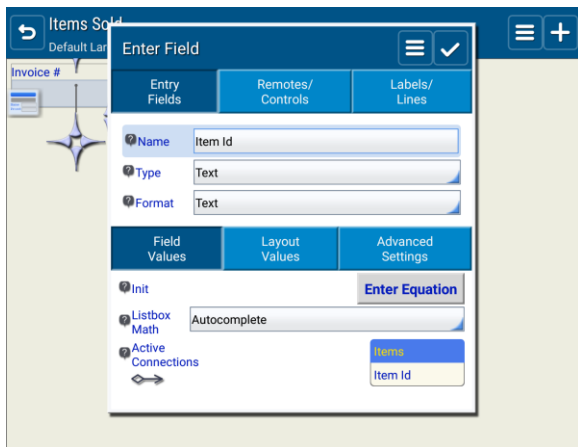


Tap **Select Field**.

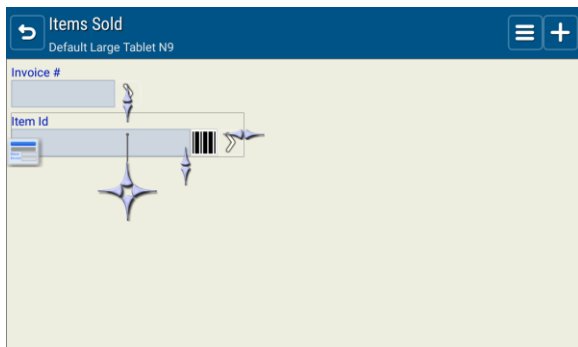
Pick the **Item Id** field.



Save the Active Connection.



Save the new *Item Id* field.



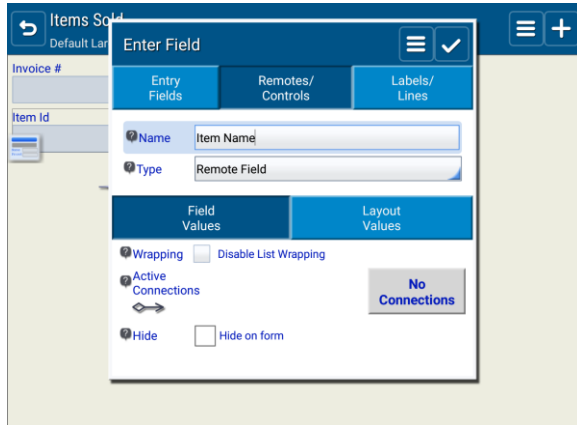
The Item Id on the Items Sold form.



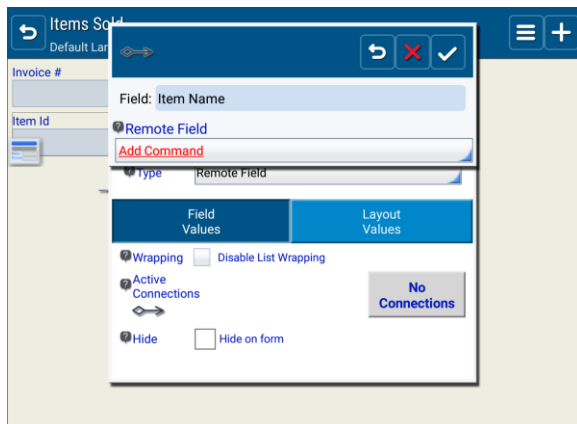
Now let's add the *Item Name*.

Now we are going to use a **Remote Field**.

Tap the **Remotes/Controls** tab.

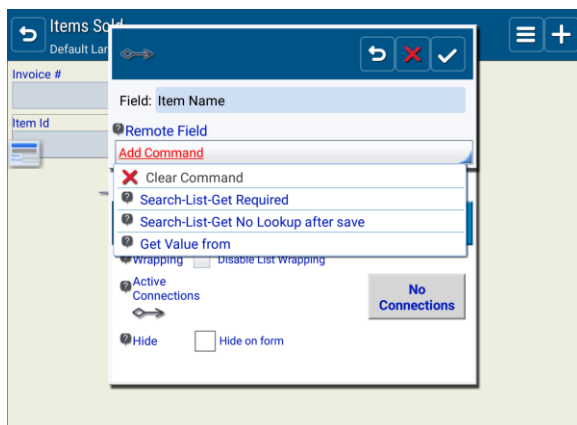


Name it *Item Name* and set the type to **Remote Field**.



Press the **No Connections** button.

Tap **Add Command**.



Pick **Search List Get No Lookup after save**.

*No lookup after save* will let us use this to lookup an Inventory Item by name and link to it, but not use this value as a lookup key after the item is saved.

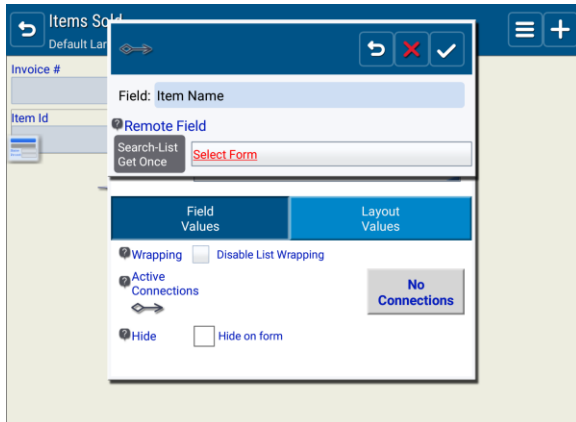
The main relation used to link to the inventory item will be the *Item Id*.

### PRO TIP:

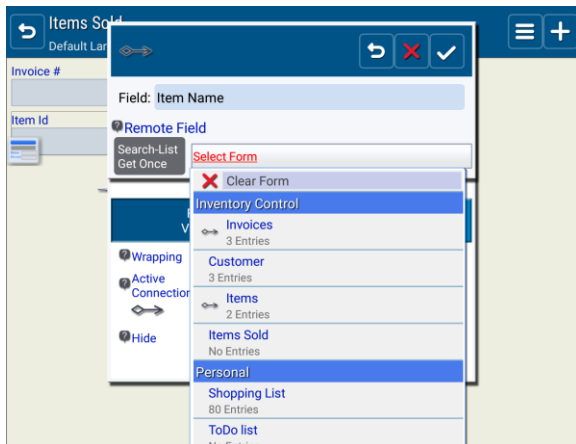
This pattern of a **Unique Key** lookup and save (Item ID), and a secondary **Remote Field** lookup (Item Name) is commonly used in **SailformsPro**. Search List Get Once is used so the item selected cannot be changed in the future. The same reason we used Write Once on the Item ID.

If you just used Search-List-Get, then the user could lookup other items after this entry is saved, which could cause inventory quantities to become invalid.

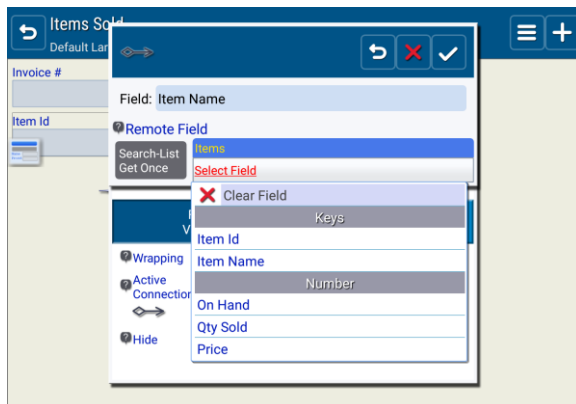
Once a you setup **Active Connections** that do **Add to's** and **Subtract from's**, the values will be kept balanced with any changes or deletes, using the **Unique Key** to lookup and change the values. If the **Unique Key** lookup is changed, the Adds and Subtracts will work against the wrong items.



Tap **Select Form**.

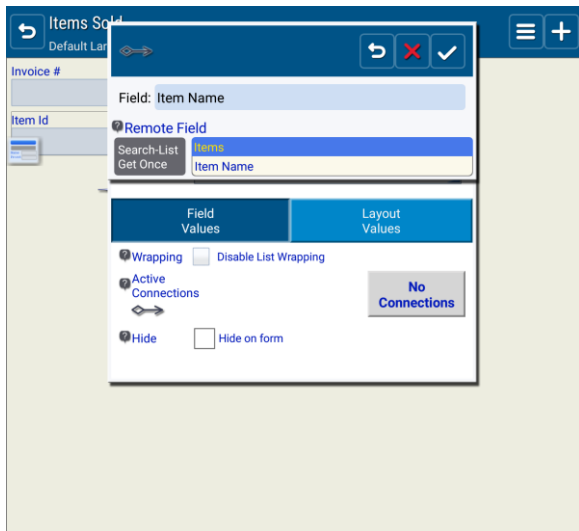


Pick the **Items** form.

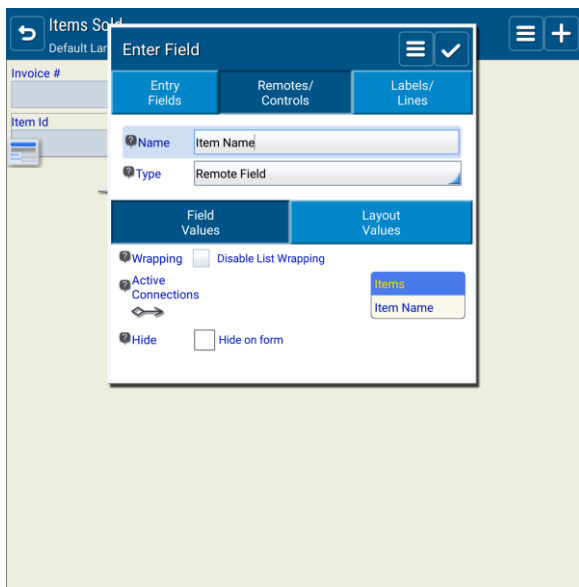


Tap **Select Field**.

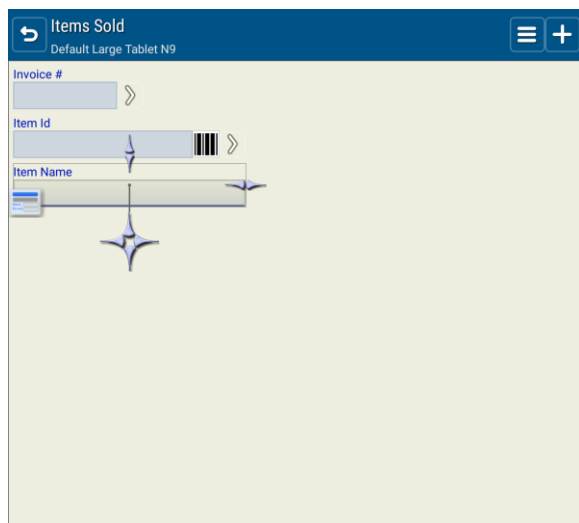
Pick the **Item Name** field.



Save the Active Connection.

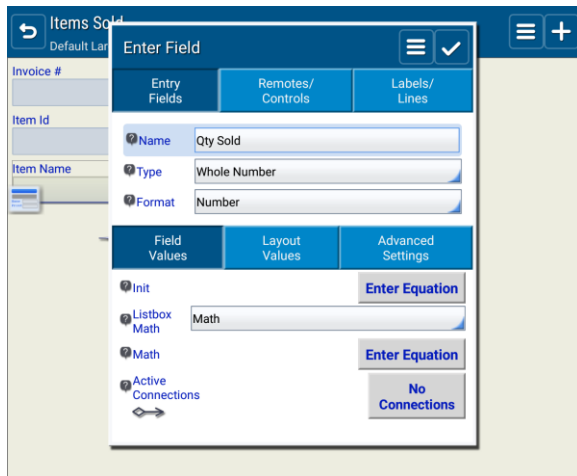


Save the new *Item Name* field.



New *Item Name* field on the Items Sold form.

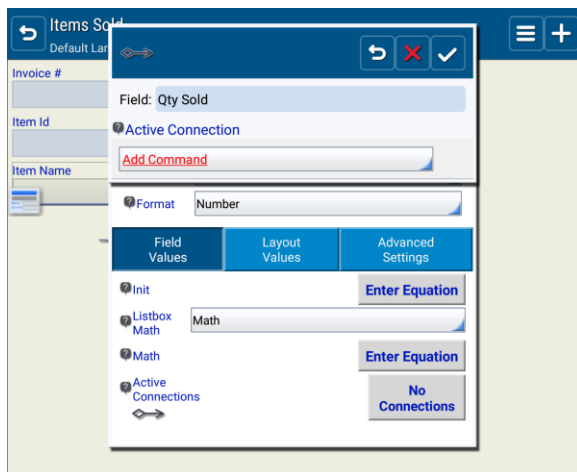
Now let's create the quantity sold field. This Active Connection will automatically subtract from the current *On Hand* amount in the Inventory Item and add to the quantity sold field.



Add an new Whole Number field and name it *Qty Sold*.

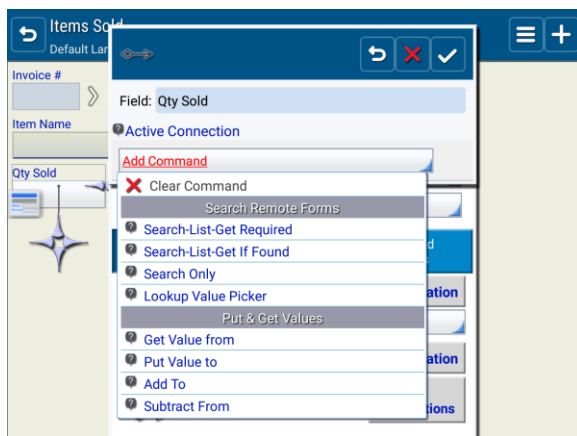
This will be used to enter the number of items sold.

Click the Advanced Settings tab and check the **Required** checkbox to make sure the quantity if filled in.

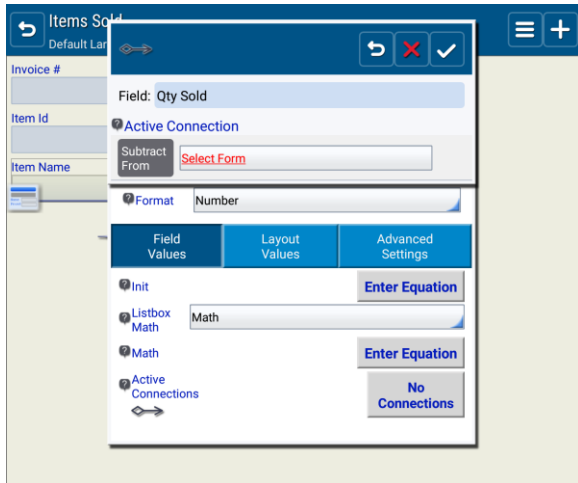


Tap the **Field Values** tab and click the Active Connection button **No Connections**.

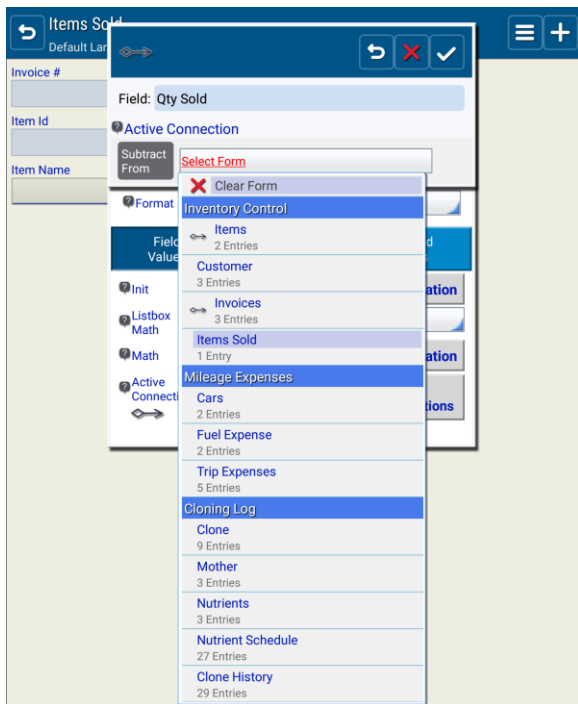
Tap on **Add Command**.



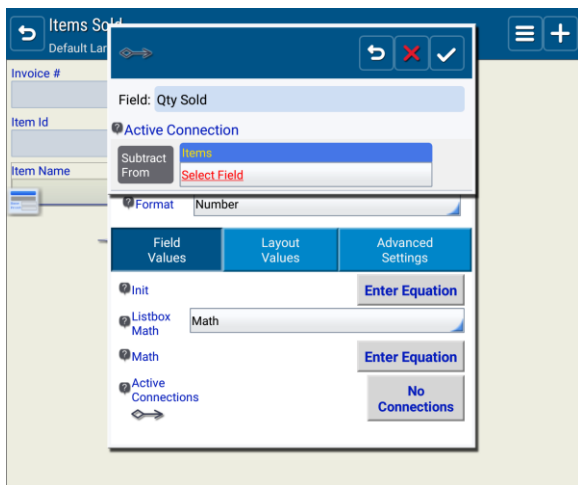
Pick **Subtract From**.



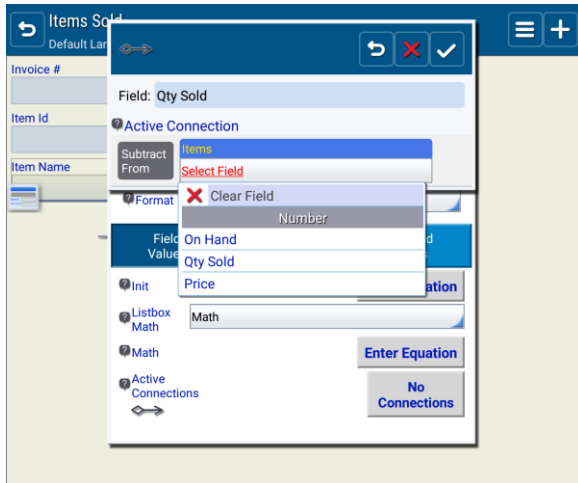
Tap on **Select Form**.



Pick the **Items** form.



Tap on **Select Field**.

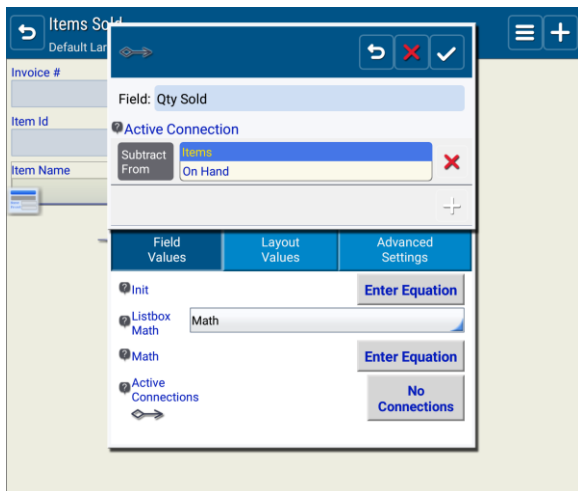


Pick the **On Hand** field in the Items form.

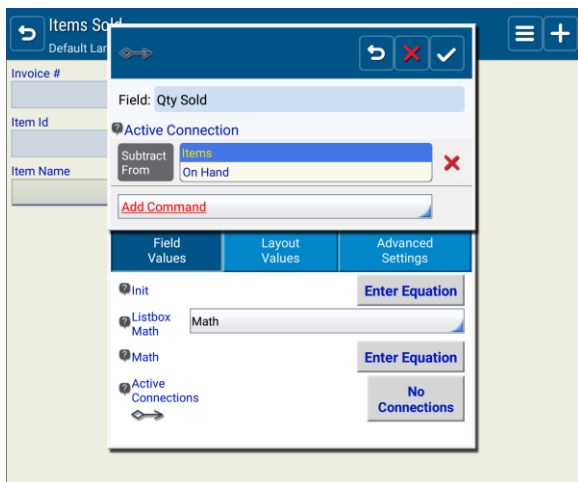
The **Subtract From** Active Connection will subtract the *Qty Sold* from the *Items On Hand* number field. If you change the *Qty Sold* amount, the amount in the *On Hand* field will automatically adjust to the change.

If you delete the entry, the *On Hand* field will automatically adjust back as if it was never sold.

Now let's add an **Add To** Active Connection to add the *Qty Sold* to the total sold number field.



Press the  $+$  to add another command.

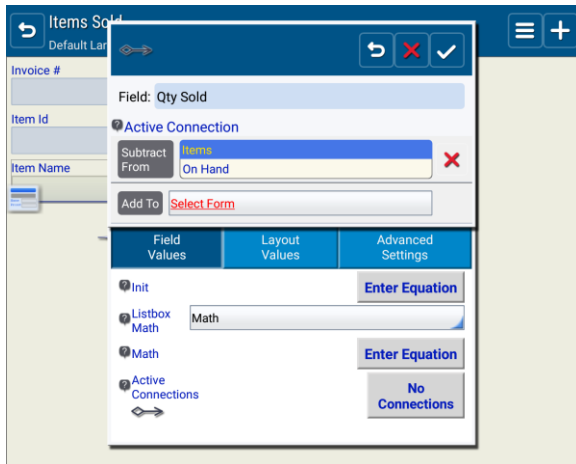


Tap on **Add Command**.

Select **Add To**.

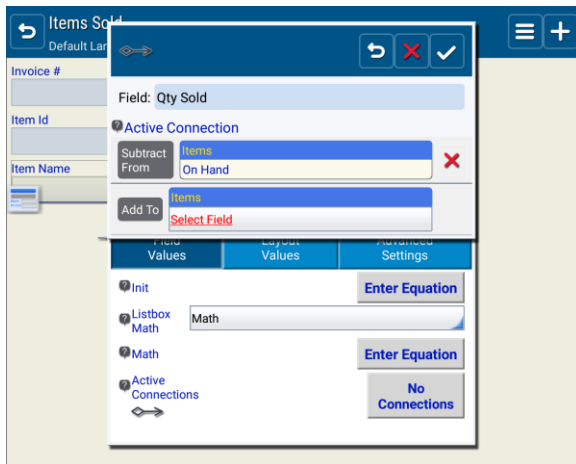
The **Add To** Active Connection will add the *Qty Sold* to the Items *Qty Sold* number field. If you change the *Qty Sold* amount, the amount in the Items *Qty Sold* field will automatically adjust to the change.

If you delete the entry, the *Qty Sold* field will automatically adjust back as if it was never sold.



Tap on **Select Form**.

Pick the **Items** form.



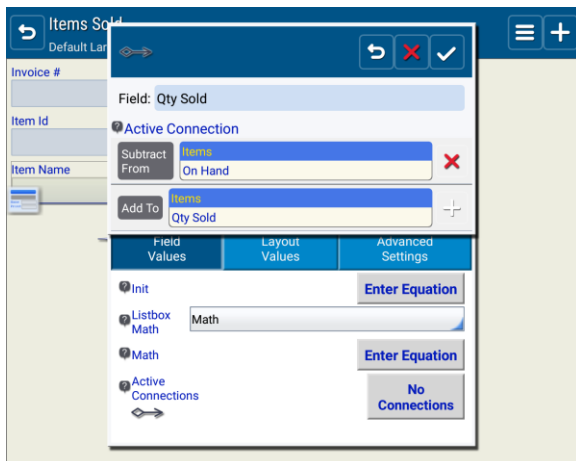
Tap on **Select Field**.

Pick the **Qty Sold** field.

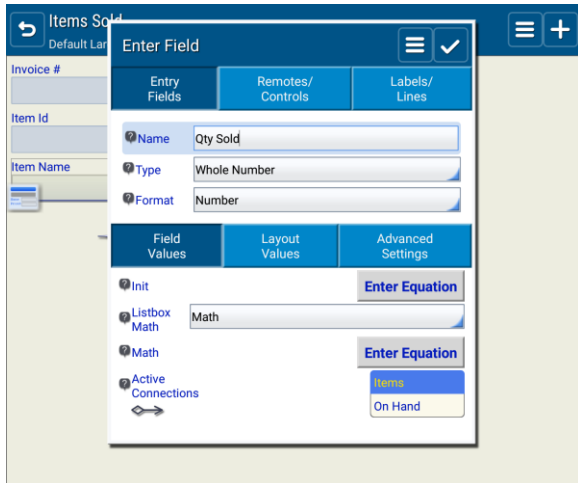
When you enter a sale, the quantity sold will automatically subtract from the item's on hand amount, and add to the item's sold amount.

If you change the amount later, both values will automatically adjust base on the changed amount, always leaving the values in balance.

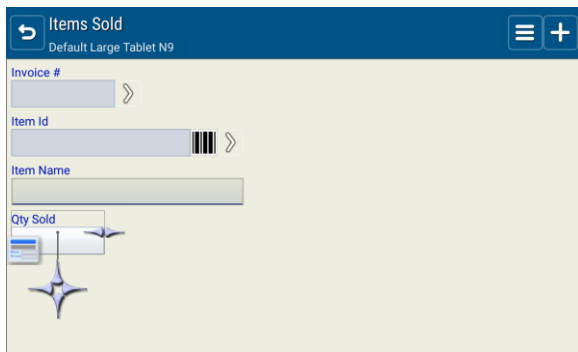
If you delete this entry, the values will be automatically reversed.



Save the Active Connection.



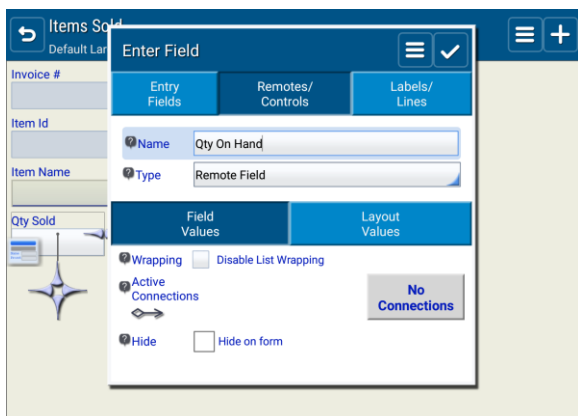
Save the *Qty Sold* field.



The form with *Qty Sold*.

When looking at your sales, it's handy to have the current Qty On Hand listed, so let's add the Items Qty on Hand field as a Remote Field.

Add a new field. Select the **Remotes/Controls** tab.

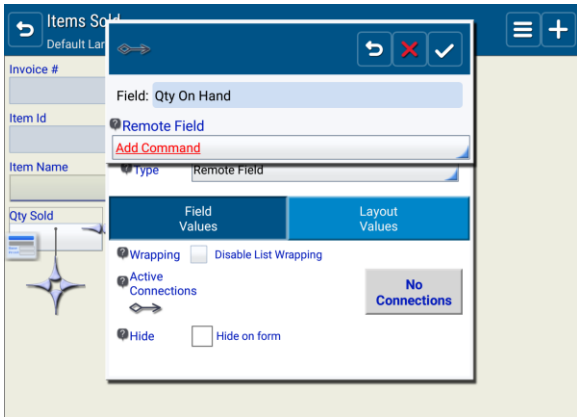


Enter *Qty On Hand* for a name.

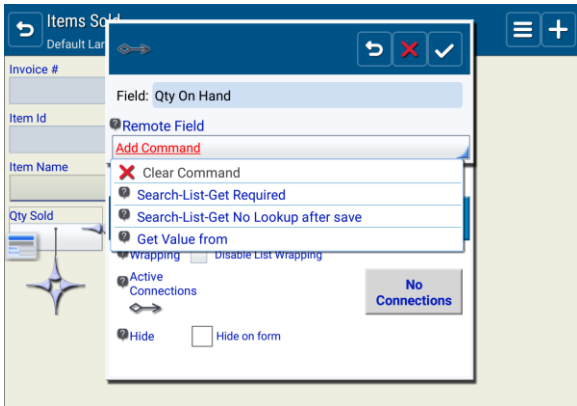
Make sure the Type is **Remote Field**.

Press the **No Connections** button.

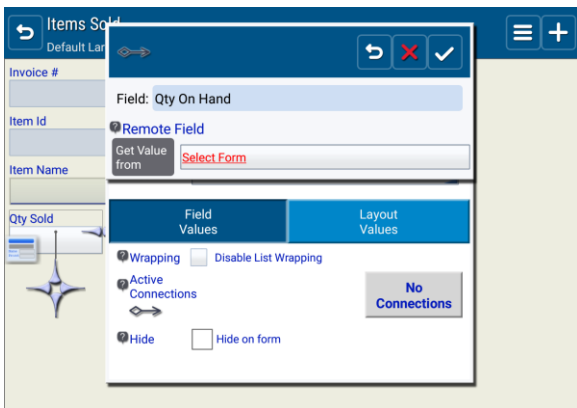




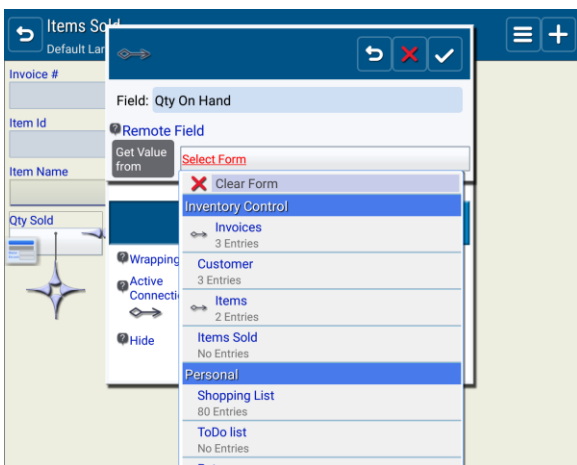
Tap on **Add Command**.



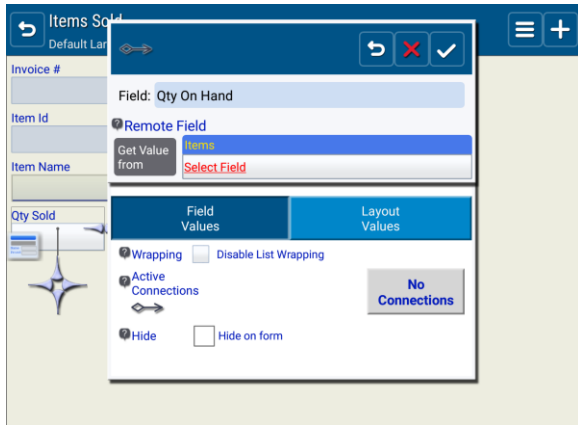
Select **Get Value From**.



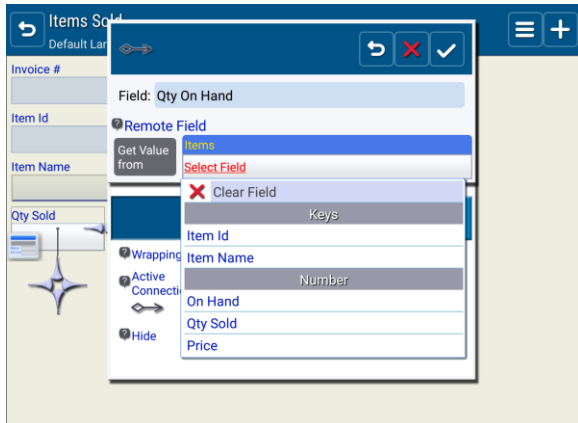
Tap on **Select From**.



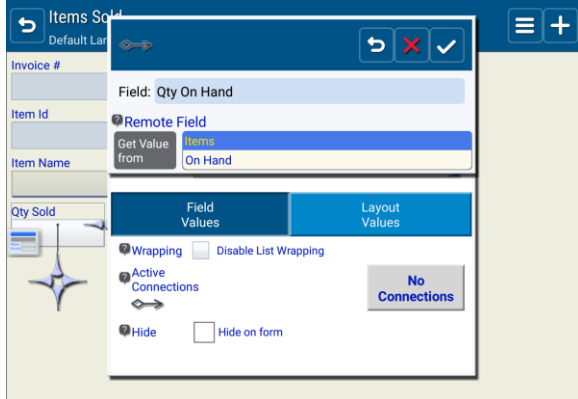
Pick the **Items** form.



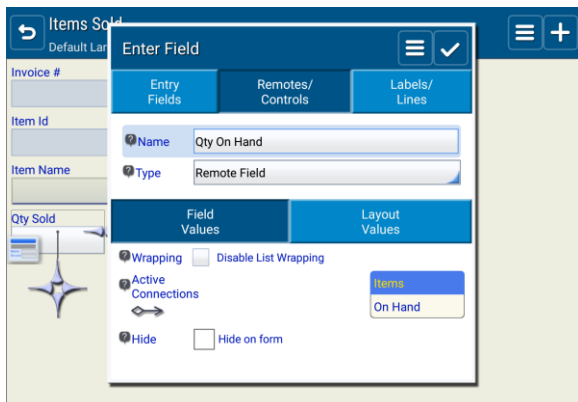
Tap on **Select Field**.



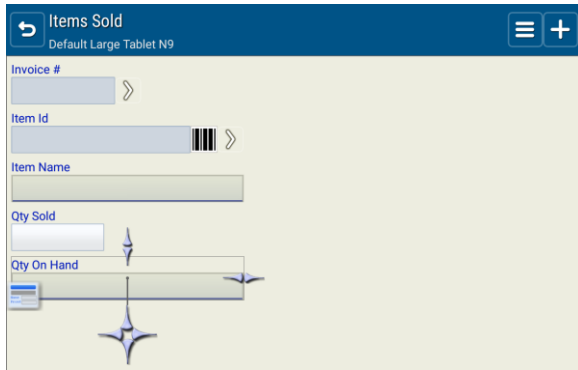
Pick the **On Hand** field.




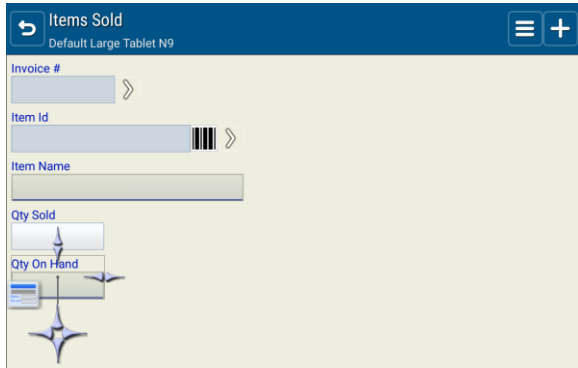
Save the Active Connection.



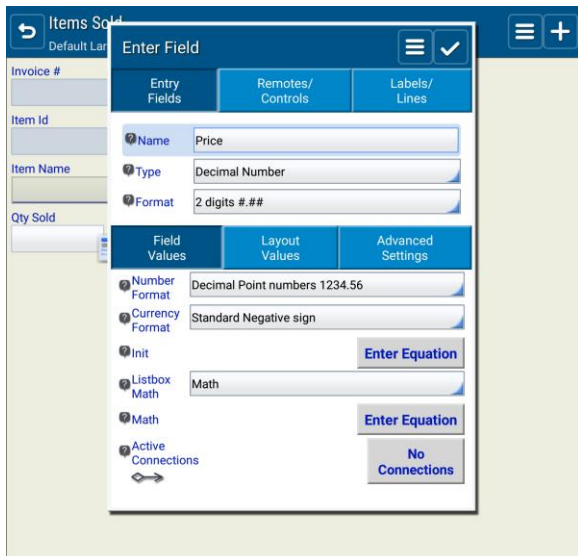
Save the *Qty On Hand* remote field.



Use the field width adjuster  to shrink the field width.

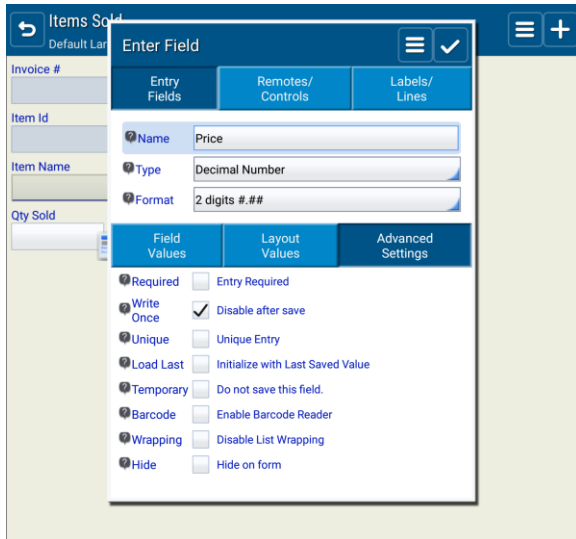


Use the dragger  to move it up to the above line.



Now create a new *Price* Field.

Set this **Type** as a Decimal Number to store the sales price of the item we are selling.



Tap on the **Advanced Settings** Tab.

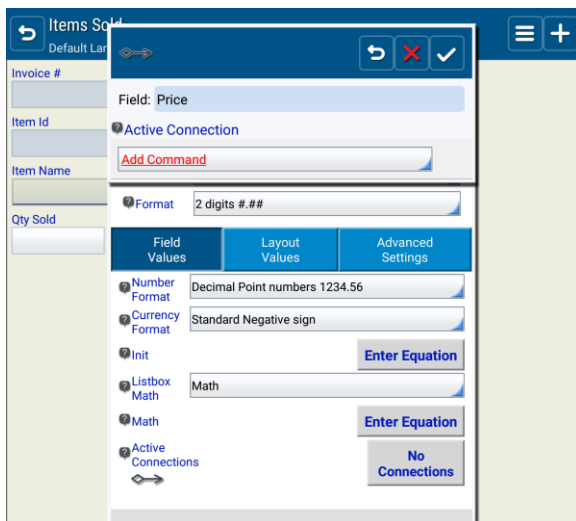
You should set this field to **Write Once**.

This will lock in the price at the time when you sold this. If you change the price of the item later, this price will not change.

If you do not set **Write Once** and allow this price to change, if you are looking at old entries, it will get the current price, then this entry will want

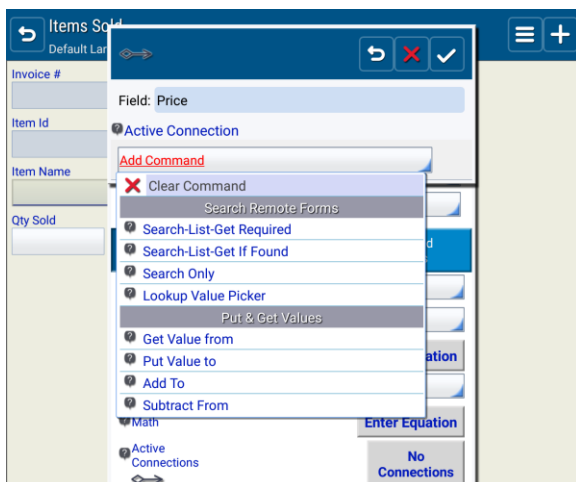
to be saved since it has changed, thus causing your Invoice amount to change, requiring it to be saved and kind of throwing off your balances.

If you sold at the wrong price, it is better to delete the entry, change the price in the inventory, then redo the entry.



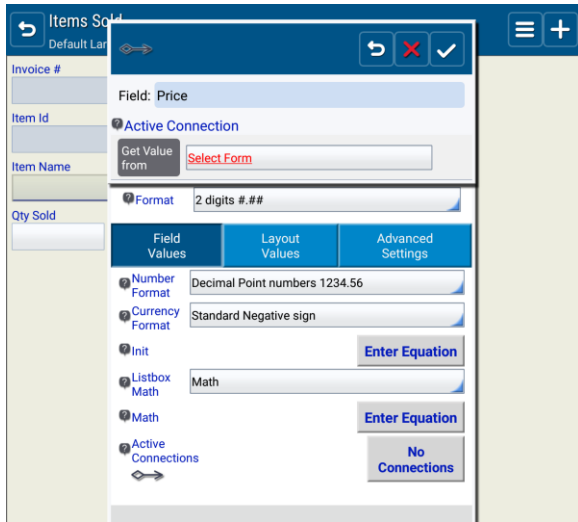
Use the **No Connections** button to add an Active Connection to get the price from the Inventory **Items** form.

Tap on **Add Command**.

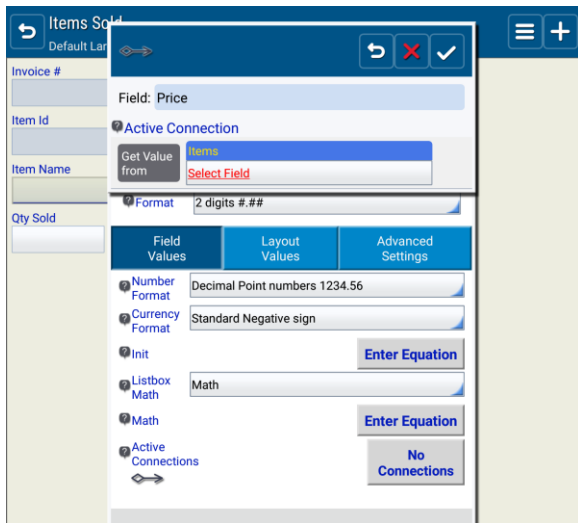


Select **Get Value from**.

This will bring over the price from the Inventory Items form.

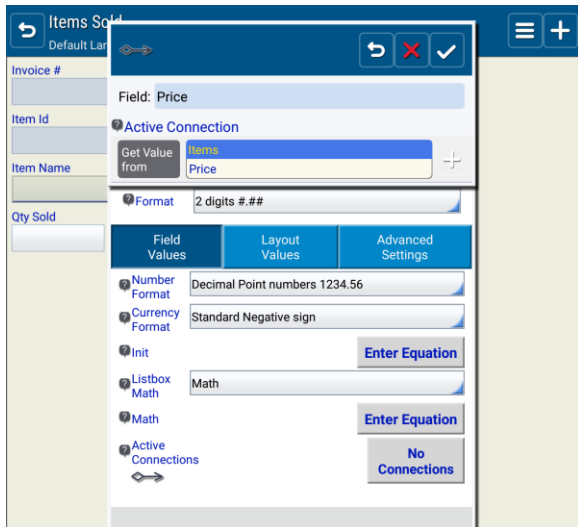


Tap on **Select From**.



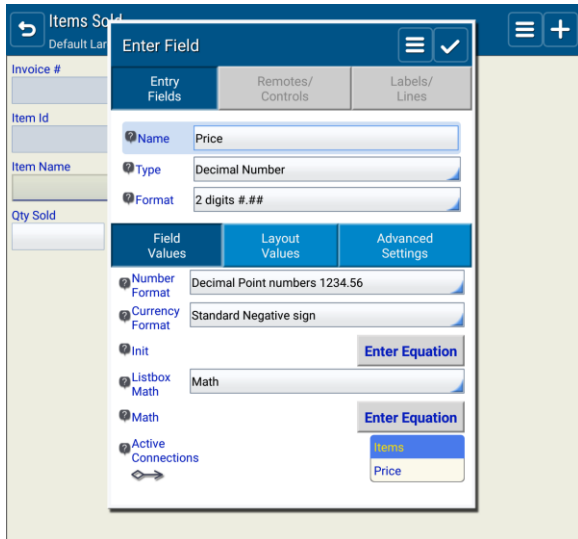
Select the **Items** form.

Tap on **Select Field**.

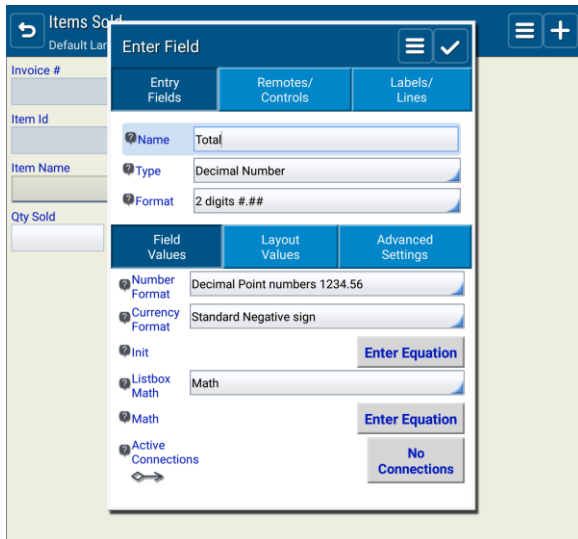


Pick the **Price** entry field.

Save the new Active Connection.



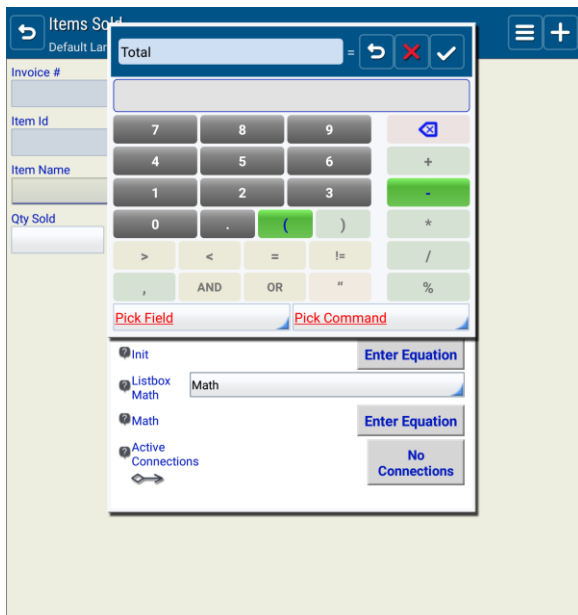
Save the new *Price* field.



Add a Total Field

This will calculate the single line item sales price.

Set the Type to **Decimal Number** and the Format to **2 digits ###**.

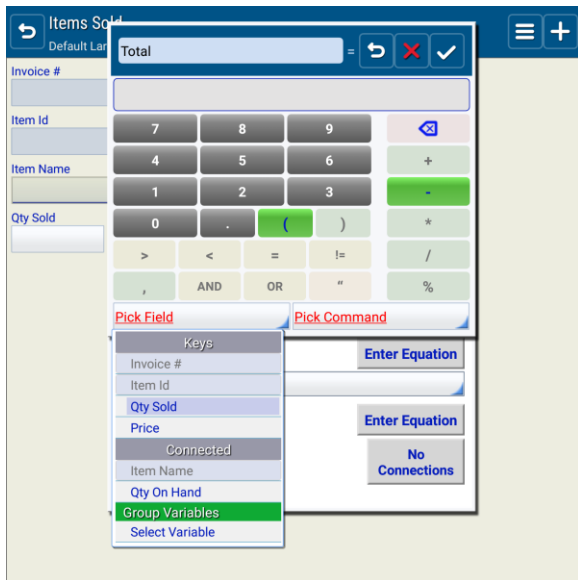


Press the **Math** button.

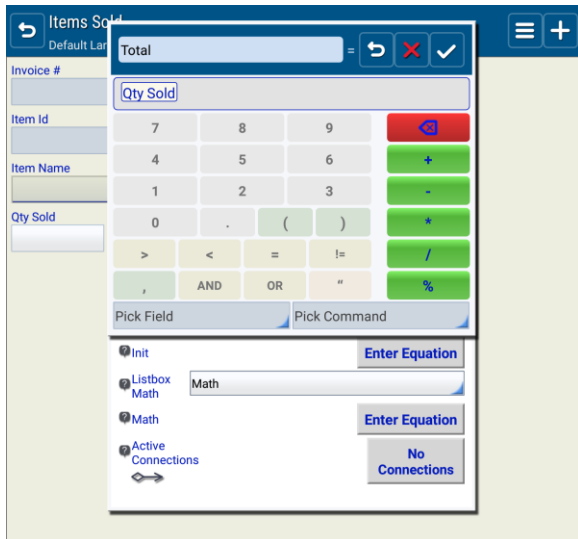
If the Math button is not displayed, press on Listbox/Math dropdown and pick **Math**.

$$\text{Total} = \text{Qty Sold} * \text{Price}$$

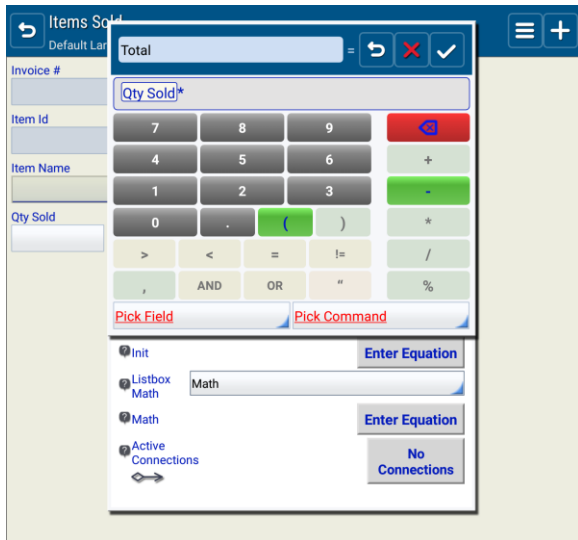
Tap on **Pick Field**.



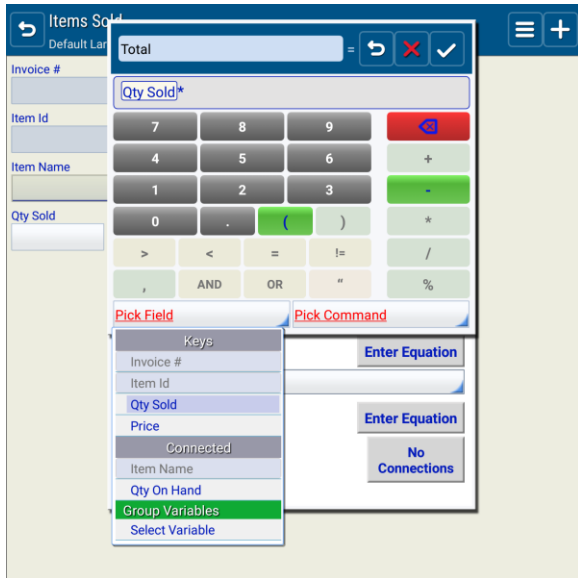
Pick the **Qty Sold** field.



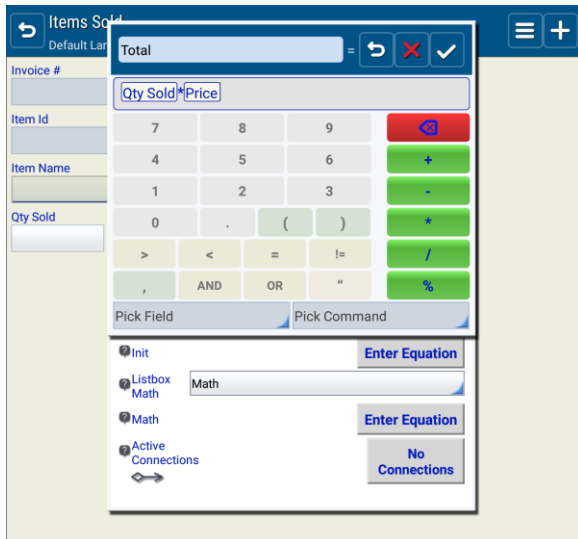
Press the Multiply Button



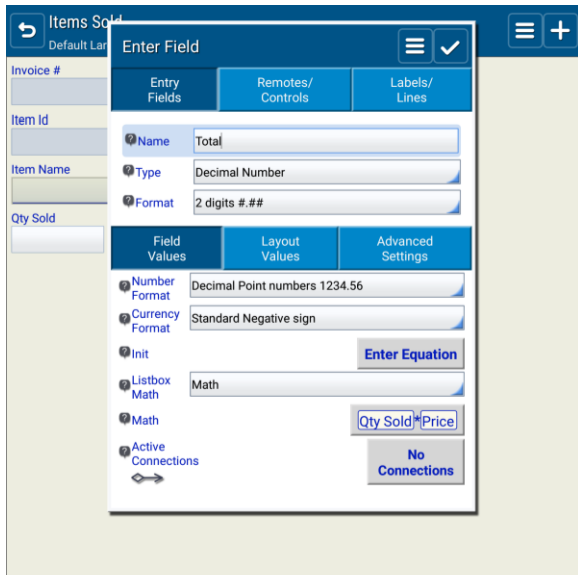
Tap on **Pick Field**.



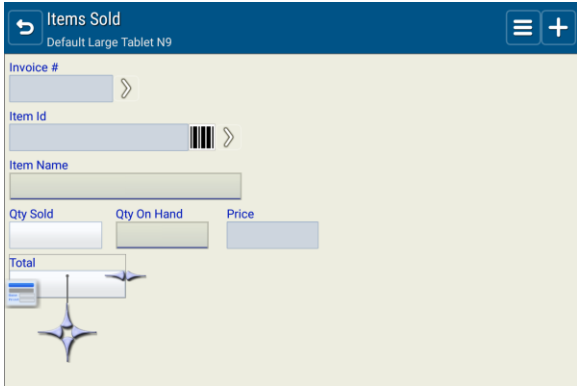
Select the **Price** field.



Save the new math equation.

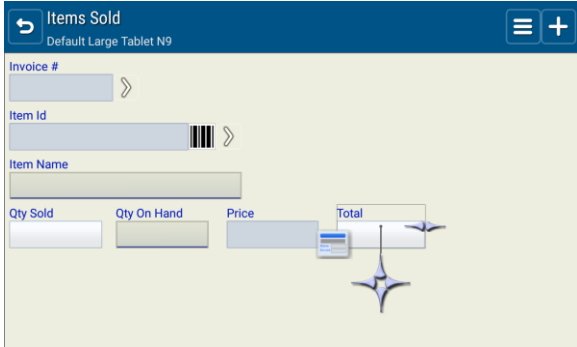





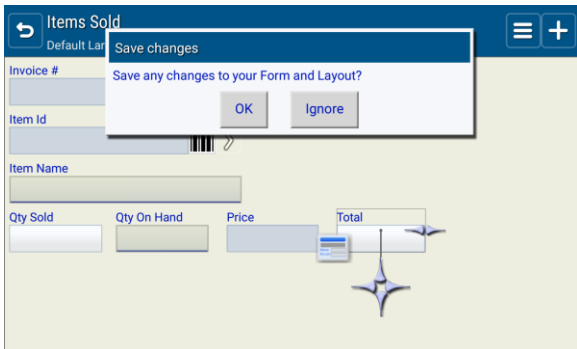


The new *Total* field.

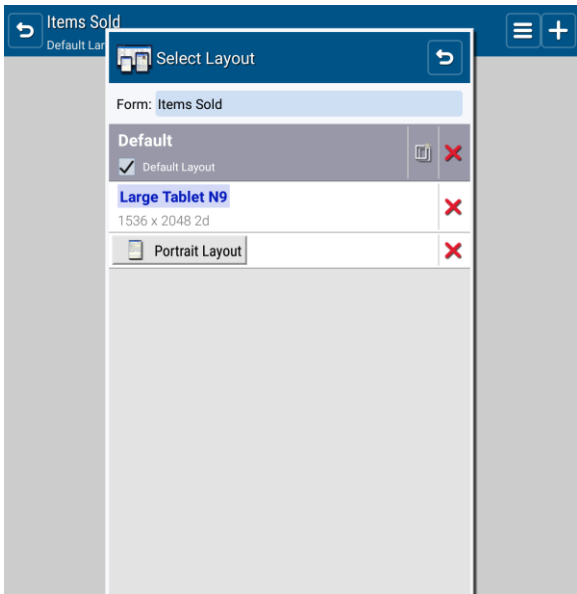
Use the move dragger to bring it up to the above line.




Press  to finish working on the Items Sold form.



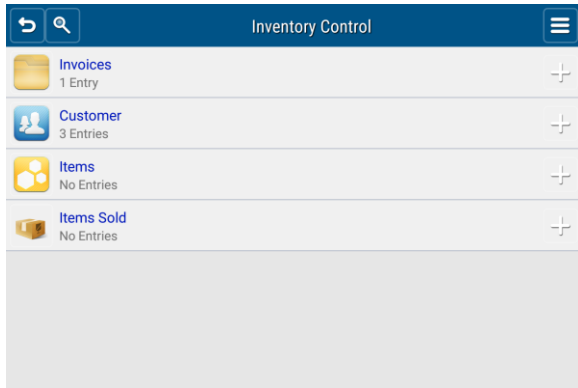
Press OK to save the changes.



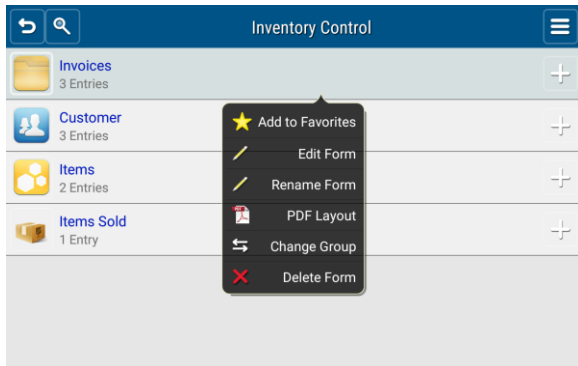
Press  to exit the Select Layout picker.

## Embedded Remote Tables

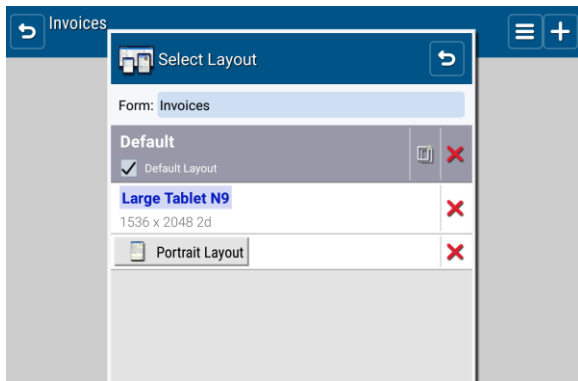
Back to our Invoice, let's add an **Embedded Remote Table** to show and add Line Items for the Invoice.



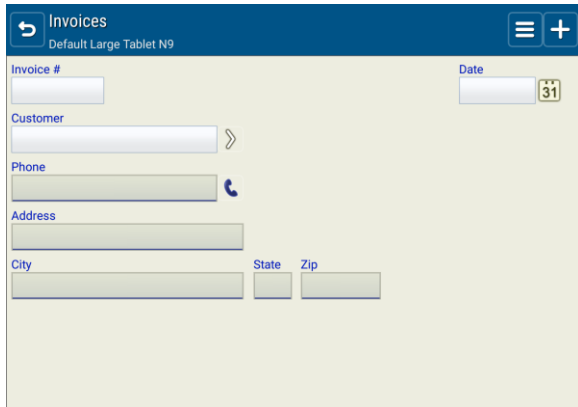
Long press or double tap on the Invoices.



Pick **Edit Form**.



Select your **Portrait Layout**.

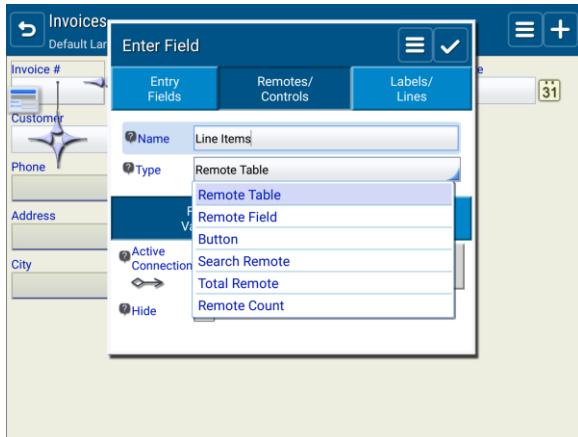


Your Invoices screen.

Now we want to add an Embedded Table to show all of the items purchased on this Invoice.

Add a new field.

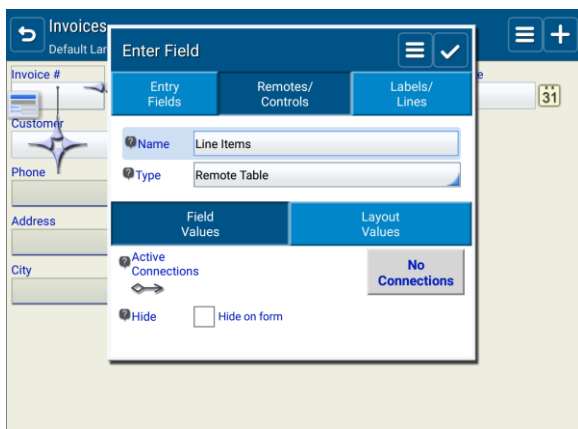
Tap the **Remotes/Controls** tab.



Enter the field name, *Line Items*.

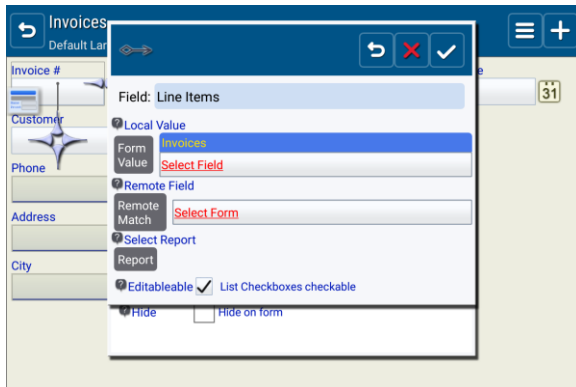
Tap the **Type** listbox and pick Remote Table.

A **Remote Table** is a remote form's List, displayed in an embedded table in the form. The entries shown are based on a field in this form (Invoices) matching a Key Value in the remote form (Line Items).



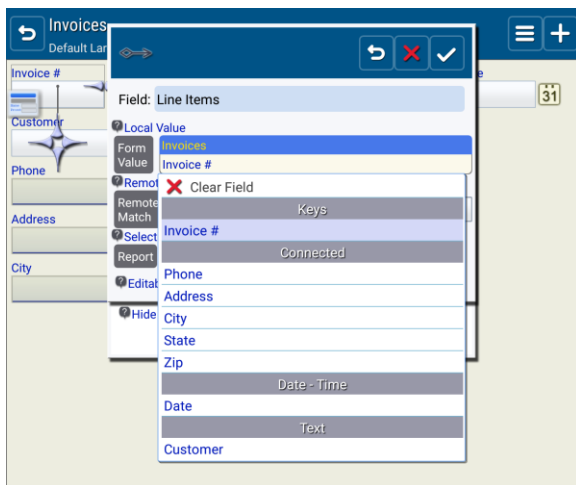
Press the **No Connections** button.

We are going to use the Invoice Number as our common matching key value.



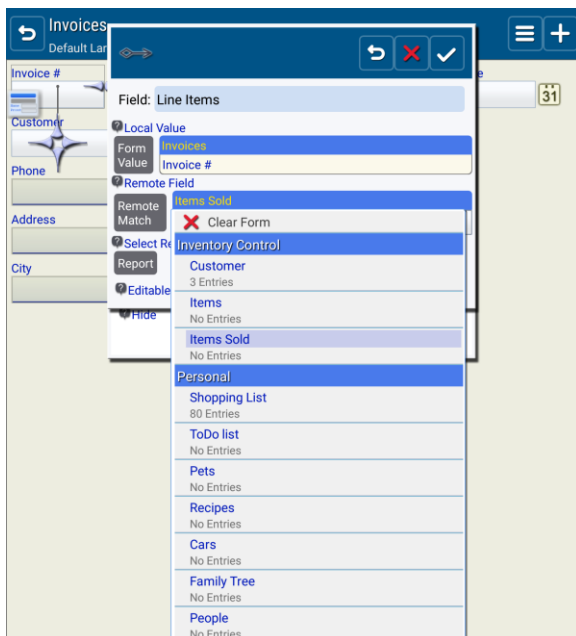
The **Local Value** is the field from the Invoice that we will use to look up the matching line items.

Tap on **Select Field**.



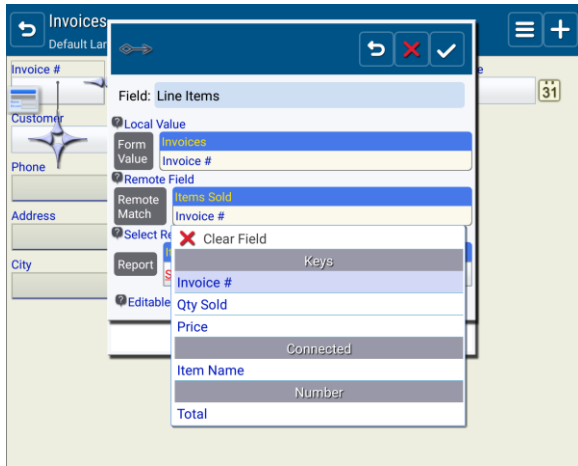
Pick the **Invoice #**.

The **Remote Field** determines which form and field we wish to match and list all of the matching values.



Tap on **Select Form**.

Select **Items Sold**.

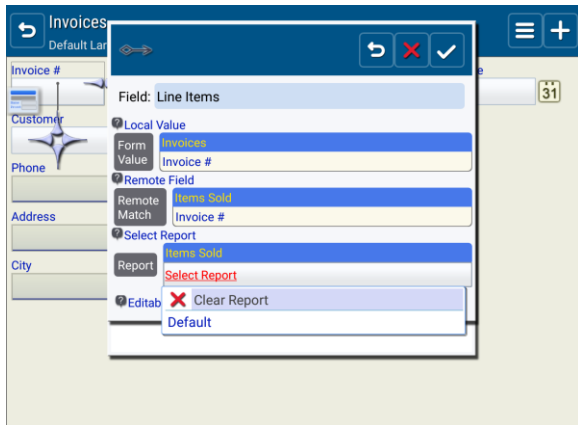


Tap on **Select Field**.

From the Items Sold form, pick the common key value field.

We are connecting the two forms with the Invoice #, so pick **Invoice #**.

Now that the connection is made, you need to pick the table's Report layout.



In Sailforms, you can save report layouts, (**Saved Lists**).

Saved lists will remember the column order, sort order, breaks, Sub-totals, totals, filters and hidden columns

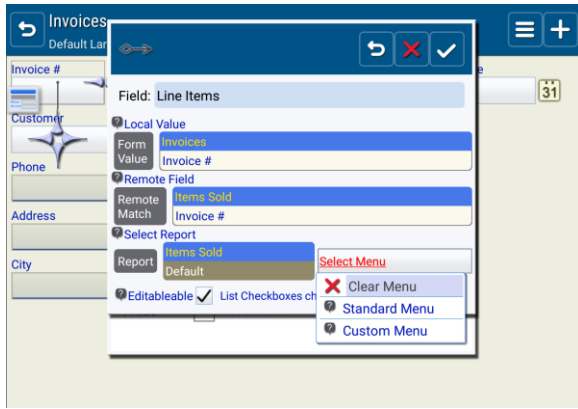
Since we have not defined a saved list yet, just pick **Default**.

That will use the current default list layout for the Items Sold form.

Changing the default layout in the Items Sold form will also change it in the embedded table, since we are using the default report.

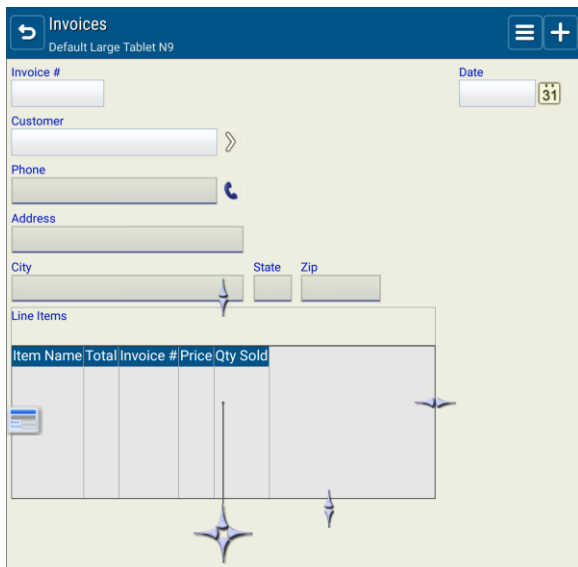
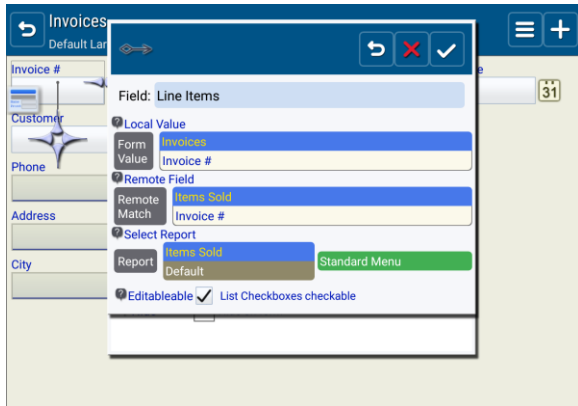
The Menu let's you can define the actions that occur when you tap on the **+** at the top right of the table.

You can also define the actions that occur when you tap on an entry in the table.



Pick the **Standard Menu**. That supports the basic actions of viewing an existing entry and adding a new entry.

See the section on **Custom Menu's** for more information.



Congratulations, you have finished the first part of setting up an Invoicing system.

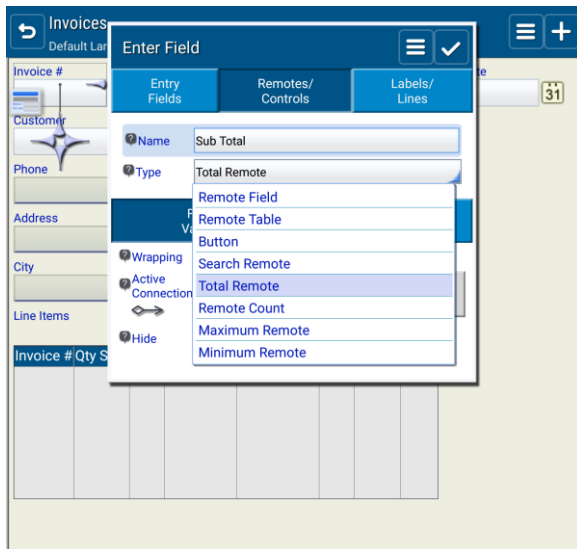
Add some Inventory Items and customers, then go sell some items with your new Invoice form.

## Remote Totals

To total the Invoice we will use the Remote Total control field.

This lets you define the common lookup key and which remote field you want to total.

Open your Invoices form to edit and create a new field.

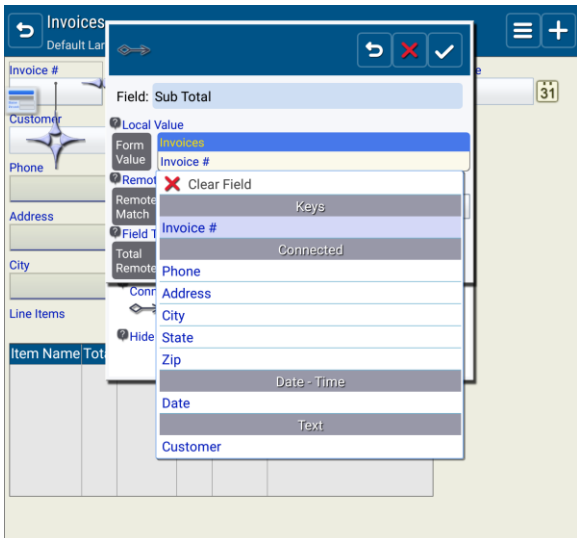


Name it *Sub Total*.

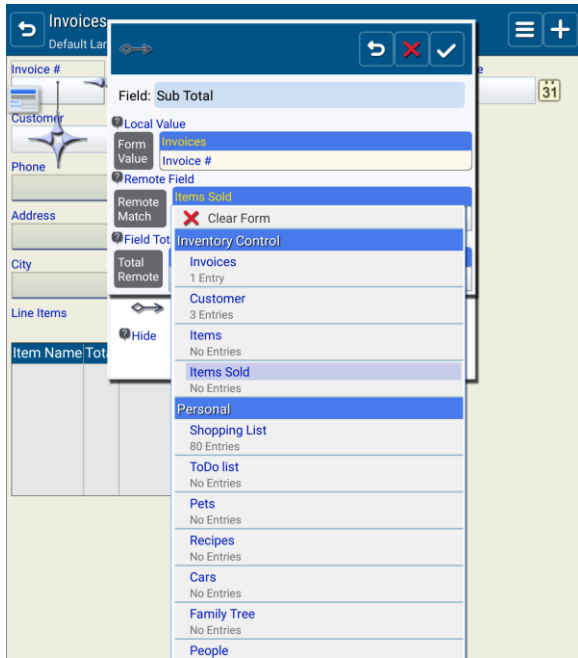
Pick **Remotes/Control** tab.

Select the Type **Total Remote**.

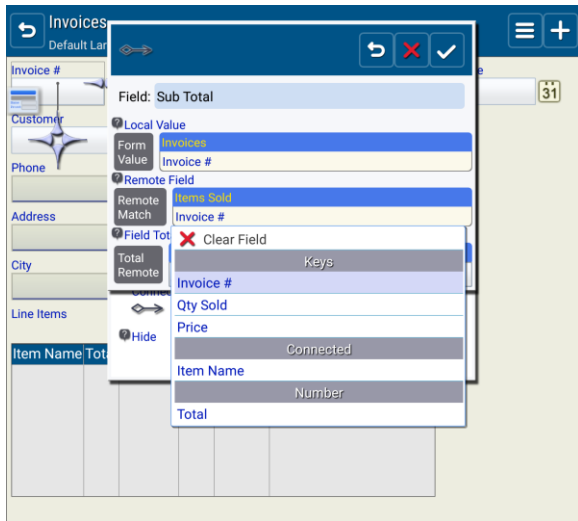
Press the **No Connections** button.



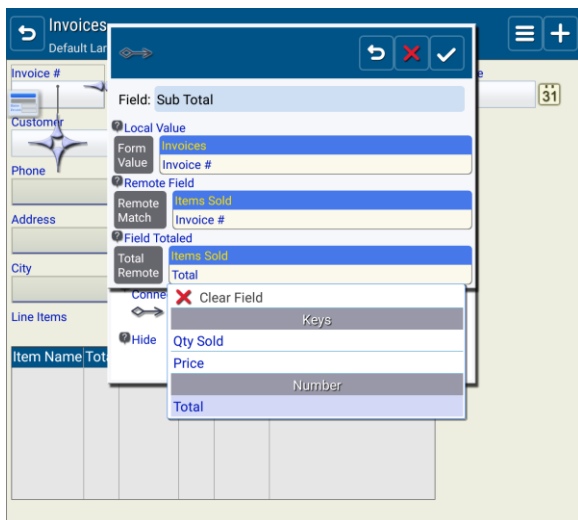
Pick **Invoice #**



Pick **Items Sold**.



Pick **Invoice #**.

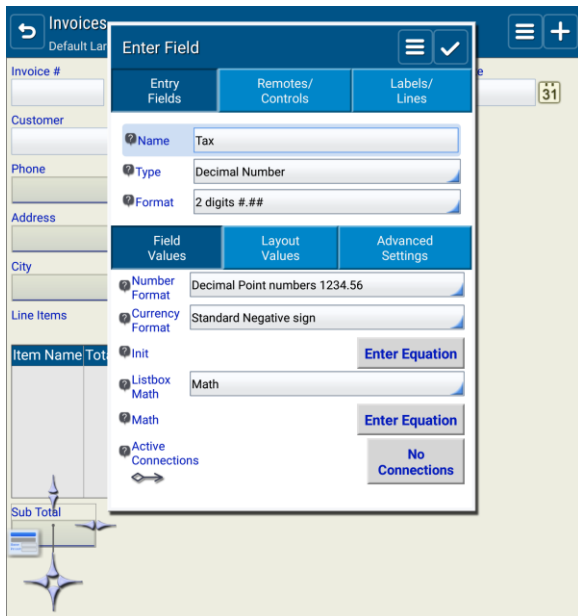
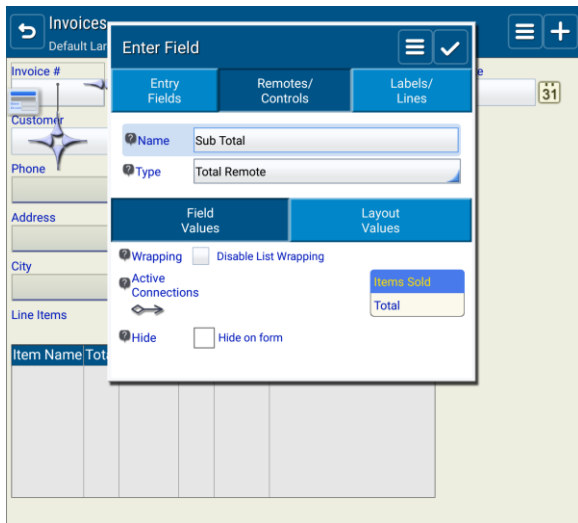
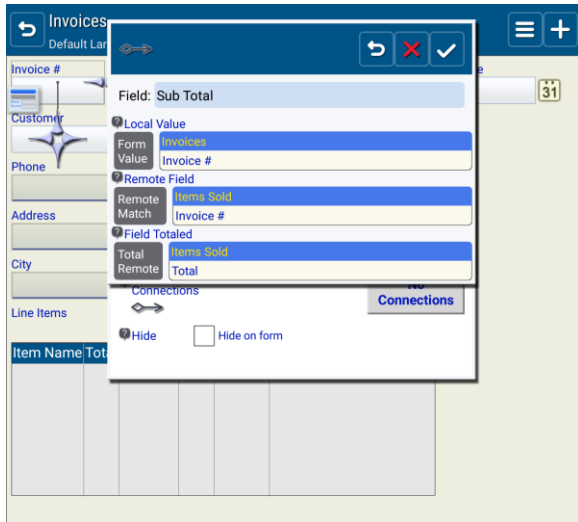


Select **Total**

This field is virtual, that is, it always totals the remote values when displayed, but the value is never saved.

To use this in a math equation and to be able to see the total value in reports, create a local field that will save the total.



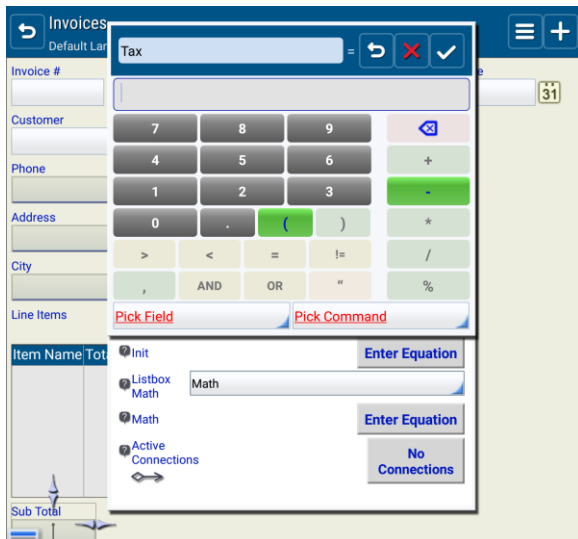


Add a Tax Field

Make this a 2 digit decimal number.

Make sure **Listbox Math** is set to Math.

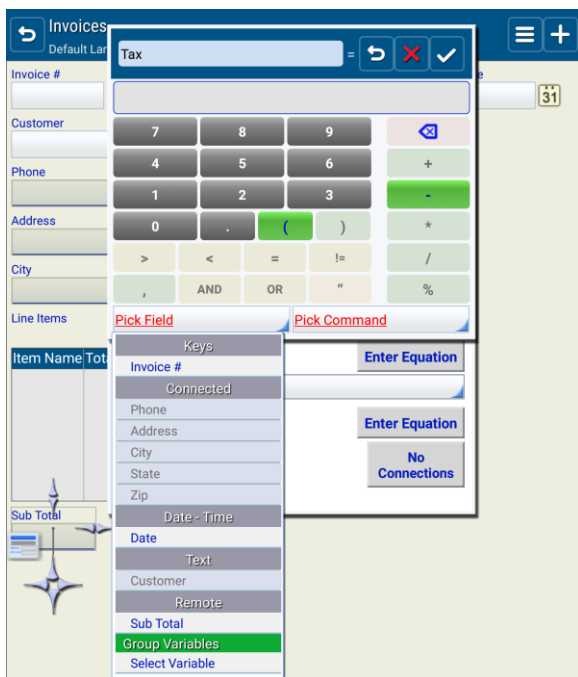
Tap the Math **Enter Equation** button.



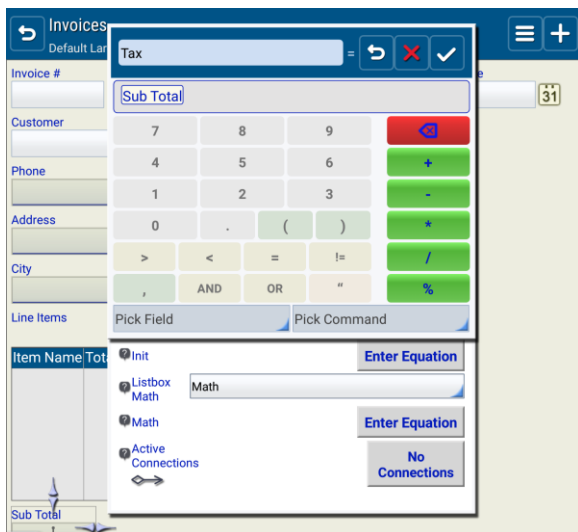
Set the equation to compute the tax.

$$\text{Tax} = \text{SubTotal} * \text{TaxRate} / 100$$

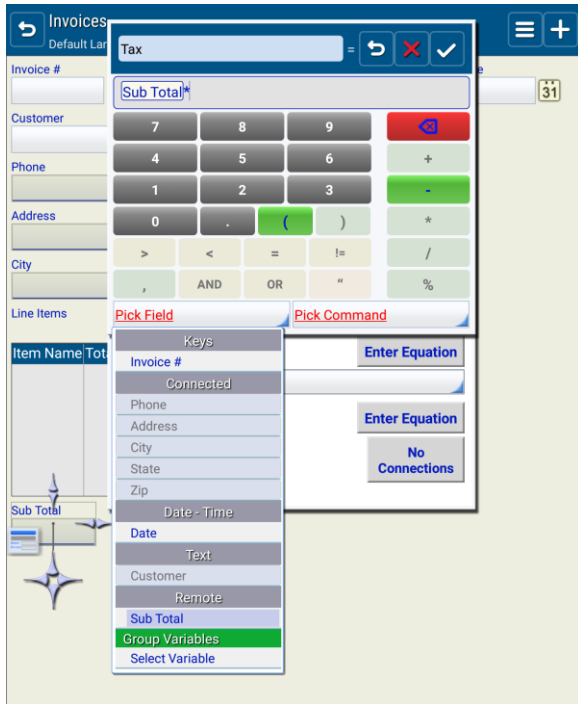
Tap **Pick Field**.



Select the **Sub Total** field.

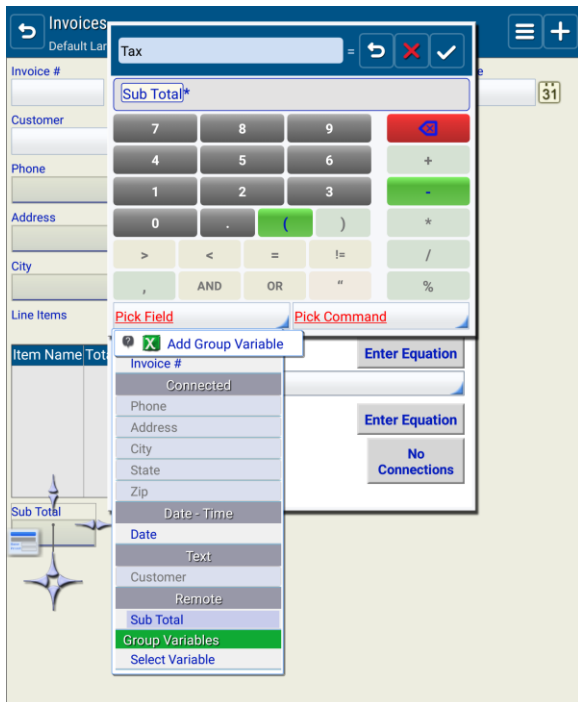


Press the  button.

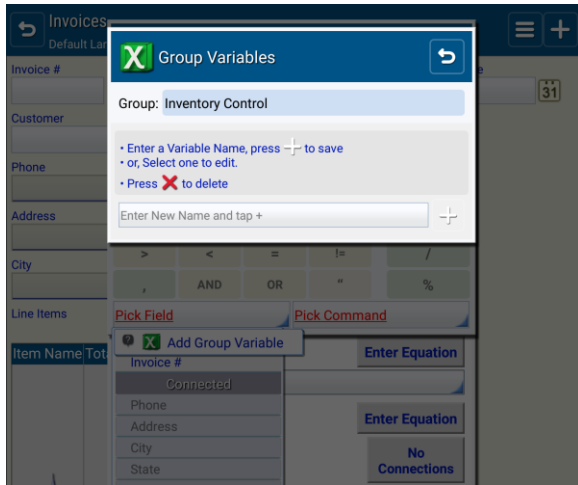


Tap **Pick Field**.

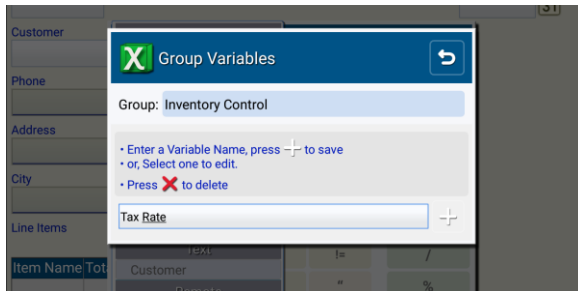
Tap **Select Variable**.



Since no Group Variables are defined, select **Add Group Variable**.

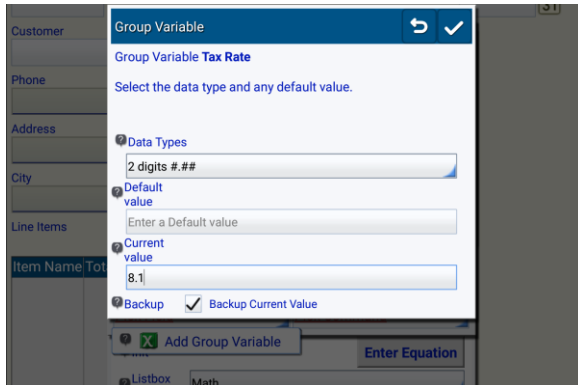


Group variables are available to Math Functions, List Filters and button commands



Enter the variable name, *Tax Rate*.

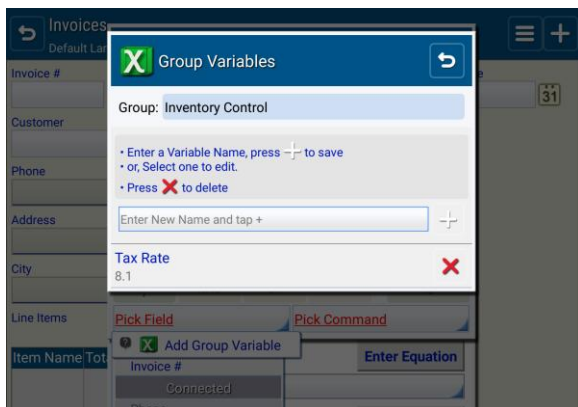
Press **+** to add the variable.



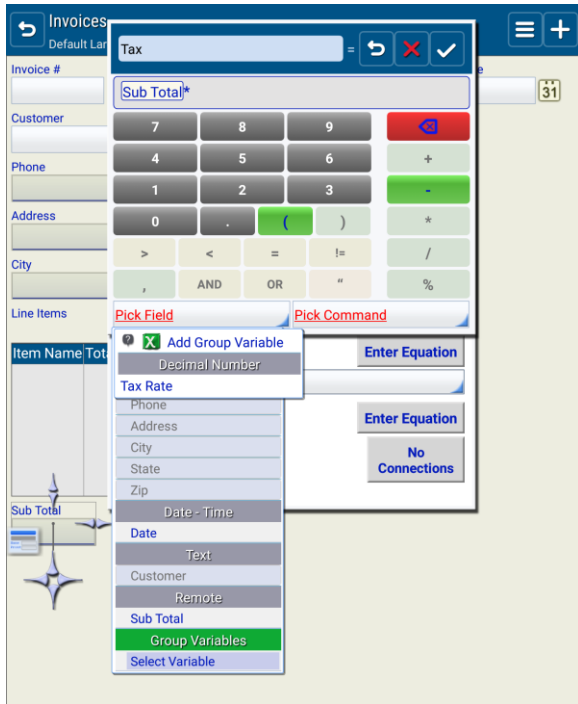
Select the Data Type 2 digits ###.

Enter your local tax rate, i.e. *8.1*.

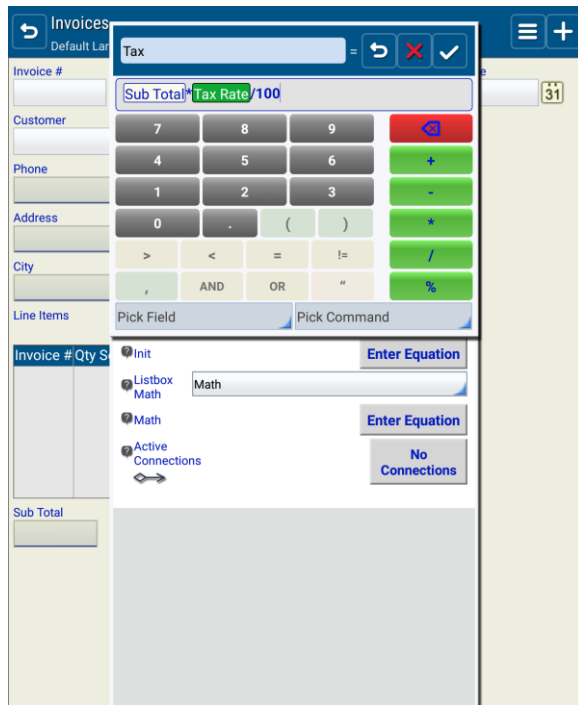
Press **✓** to save the variable.



Press **↶** to exit back.




Press **Select Variable** and pick your new **Tax Rate** variable.




Now divide by 100.

Press / and type 100

Press  to save the math equation.

The screenshot shows the 'Enter Field' dialog box with the following configuration:

- Name:** Tax
- Type:** Decimal Number
- Format:** 2 digits ###
- Field Values:**
  - Number Format: Decimal Point numbers 1234.56
  - Currency Format: Standard Negative sign
- Math:** Sub Total \* Tax Rate / 100
- Buttons:** Enter Equation (highlighted), No Connections

Press  to save the new *Tax* field.

The screenshot shows the 'Enter Field' dialog box with the following configuration:

- Name:** Total
- Type:** Decimal Number
- Format:** 2 digits ###
- Field Values:**
  - Number Format: Decimal Point numbers 1234.56
  - Currency Format: Standard Negative sign
- Math:** Math
- Buttons:** Enter Equation (highlighted), No Connections

Add a new field to calculate the Total.

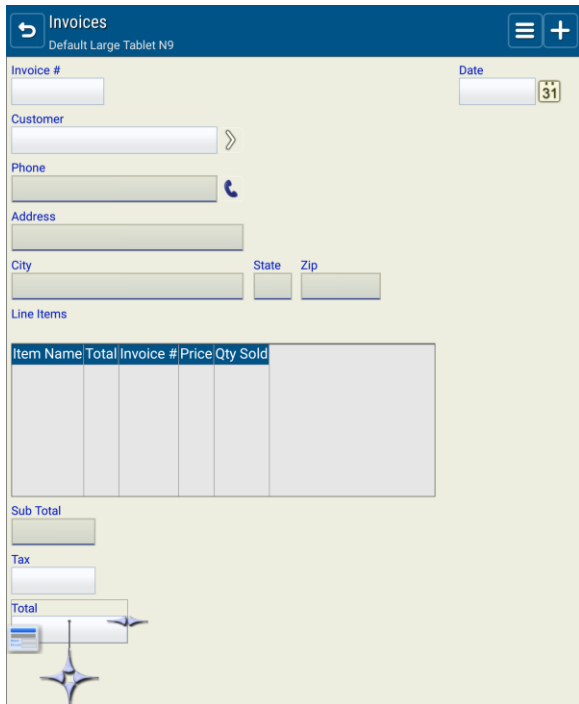
Set the name *Total* and set the type to Decimal Number.

Press the Math **Enter Equation**.



Set the Math equation

$$\text{Total} = \text{SubTotal} + \text{Tax}$$



Here is our Invoice with SubTotal, Tax and Total fields added.

## Remotes/Controls

Remote Field	<p><b>Search-List-Get</b> and <b>Search-List-Get No Lookup after save.</b></p> <p>Search-List-Get let's you search for values in remote forms to select the record, but not maintain a connection link. This is useful to find lookup remote records by a secondary key, say a phone number, name or description. Used in conjunction with an Entry Field that does a <b>Search-List-Get</b>, it will automatically fill in that fields value. When used with a keyed value that you do not want changed after the record is saved, select <b>Search-List-Get No Lookup after save.</b> That disables the ability to Lookup and change the linked value to another entry which could cause accounting errors in Linked forms.</p> <p>If you select <b>Search-List-Get</b> then you can lookup and change the value after the entry has been saved.</p> <p><b>Get value from</b> other remote forms that have been connected via other Search-List-Get fields that have established connections.</p> <p>Every time this entry is loaded, the current value of the remote entry is loaded.</p>
Remote Table	<p>Select Local and Remote Fields            Insert Remote Form List            Configure Menus</p>
Search Remote	<p>Select Local and Remote Fields. The Remote Searched field can be in the same form. This let's you recursively get values from other entries in the same form.</p> <p>Get a value from the selected field.</p> <p>No connection link is established and the value is not saved. It can be used in math equations.</p>
Total Remote	<p>Select Local and Remote Fields            Total Remote Field Value.</p> <p>No connection link is established and the value is not saved. It can be used in math equations.</p>
Remote Count	<p>Select Local and Remote Fields            Returns the number of entries found.</p> <p>No connection link is established and the value is not saved. It can be used in math equations.</p>



Maximum Remote	<p>Select Local and Remote Fields Returns the Maximum value found.</p> <p>No connection link is established and the value is not saved. It can be used in math equations.</p>
Minimum Remote	<p>Select Local and Remote Fields Returns the Minimum value found.</p> <p>No connection link is established and the value is not saved. It can be used in math equations.</p>
Button	Execute commands when pressed.

## Button and Menu Commands

Button Commands allow you to program actions that occur when you press a button, tap on a list item or process a list of values.

Menus are predefined buttons with button commands that you can change or remove.

The **Standard Menu** choice assigns the default commands available when viewing a list of entries, entering a new entry or viewing and changing an existing entry.

Lists and Embedded tables have two commands for the embedded tables, Add a new entry and show an entry.

Commands can also be assigned to buttons that you add to a form.

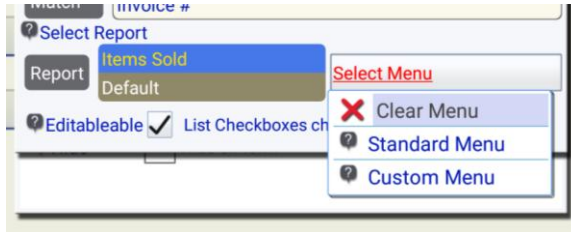
The available commands vary based on what actions can be performed for the state you are in. The four states are:

- **Button Commands**  
When an Entry Form is displayed or when showing a new form.
- **List**  
When displaying a list of form entries.
- **Table**  
Used when tapping on an Embedded table Add button or tapping on an entry in the Embedded table.
- **List Process**  
Commands available when defining a Process to execute on a list of entries.

## Standard Menu Values

There are three Standard Menus.

- Form Entry Menu
- List display Menu
- Embedded Table Menu



### PRO TIP:

When the menu choice Select Menu appears, you can select **Custom Menu** to view what the Standard Menu commands are.

## Form Entry Menu

Go to Form

Go to form is used to display a form, with an entry or cleared for a new entry.

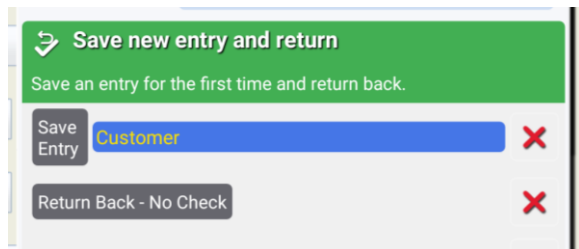
## Form Menu



## Save new entry and return

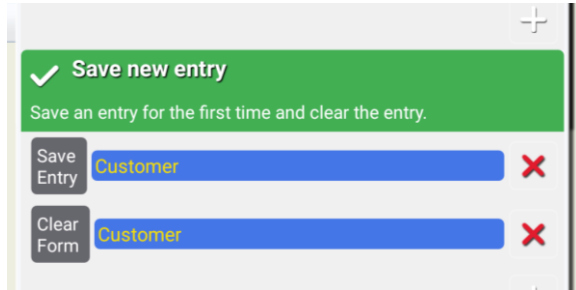
This is executed when you press **Save** when adding a new entry.

The Save and Replace commands are exclusive. Only one of the two are shown depending if you are adding a new entry or editing an existing entry.



Save Entry	Save the new entry if valid
Return Back - No Check	Go back without checking if it has changed.

## Save new entry

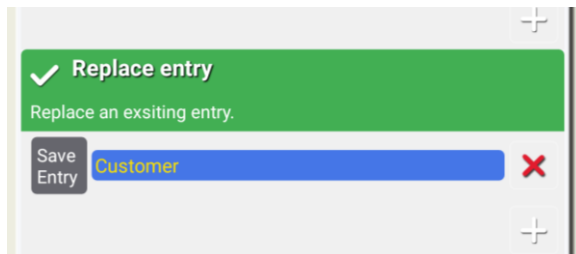


This is executed when you press **Save** when adding a new entry.

The Save and Replace commands are exclusive. Only one of the two are shown depending if you are adding a new entry or editing an existing entry.

Save Entry	Save the new entry if valid
Clear Form	Clear the entry and prepare for a new entry

## Replace entry



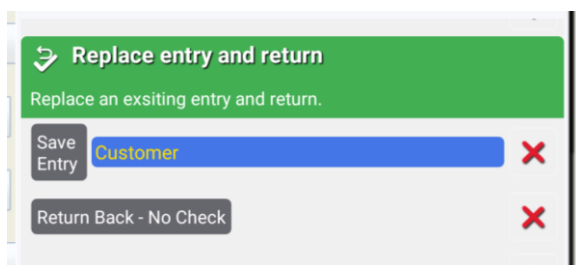
This is executed when you press **Replace** to update an existing entry.

The Save and Replace commands are exclusive. Only one of the two is shown depending if you are adding a new entry or editing an existing

entry.

Save Entry	Replace the entry if valid
------------	----------------------------

## Replace entry and return

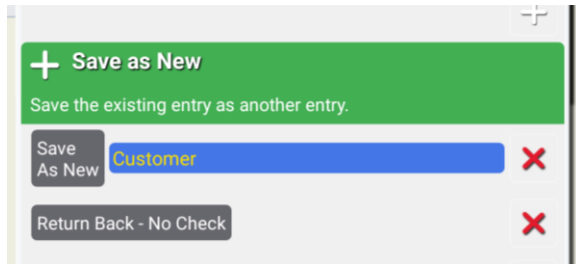


This is executed when you press **Replace** to update an existing entry.

The Save and Replace commands are exclusive. Only one of the two is shown depending if you are adding a new entry or editing an existing entry.

Save Entry	Replace the entry if valid
Return Back - No Check	Go back without checking if it has changed.

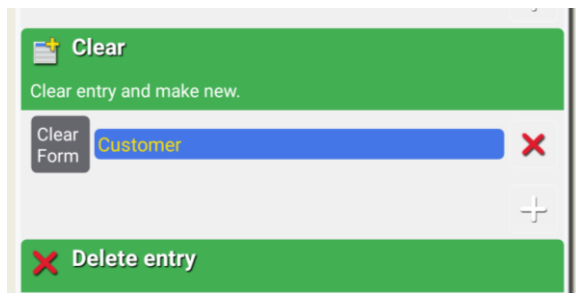
## Save as New



Saves a current entry as a new entry. Enabled only when an existing entry is shown.

Save As New	If an existing entry, save as a new entry.
Return Back - No Check	Go back without checking if it has changed.

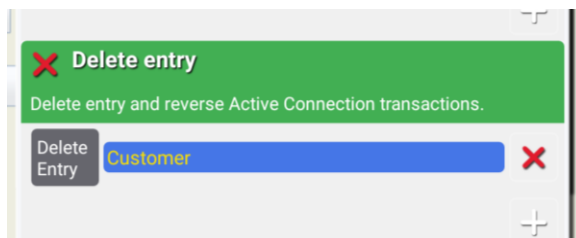
## Clear



Executed when clearing an entry. Enabled only when an existing entry is shown.

Clear Form	Clear the entry and prepare it like it is a new entry.
------------	--

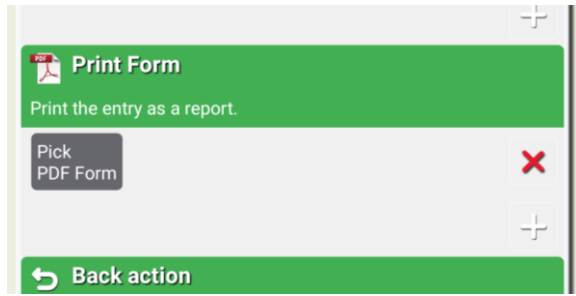
## Delete entry



Executed when deleting an entry. Enabled only when an existing entry is shown.

Delete Entry	Delete the record. Return back if no other entries can be shown.
--------------	--

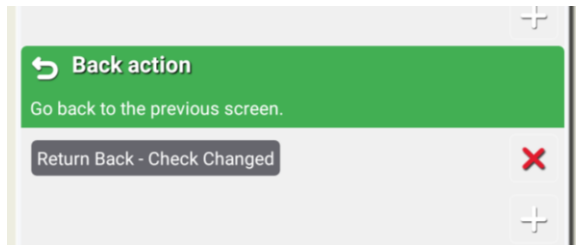
## Print Form



Pick PDF Form

Show the PDF layouts to pick one to list.

## Back action



Executed when you press Back or select the back menu choice.

Return Back - Check Changed

The action when the user presses back. Check to see if the entry has changed. Return back if not changed, otherwise show the Save Changes or Ignore question.

Back action

## Back action

Return Back - No Check

The action when the user presses back. Return back to the last screen.

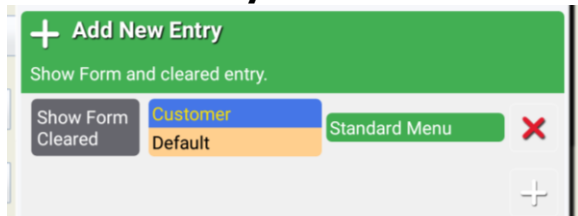
## Go to List



Go to List is used to display a form's entries in columnar form.

Columns are ordered, sorted, filtered, hidden, totaled and sub-totaled, depending on the Saved List or default list.

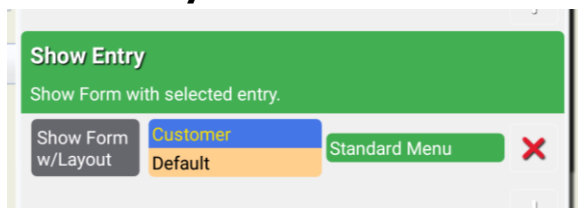
## Add New entry



Executed when showing a list of entries and the user press +

Show Form Cleared	Show the Form and clear the entry ready for a new entry.
-------------------	--

## Show Entry

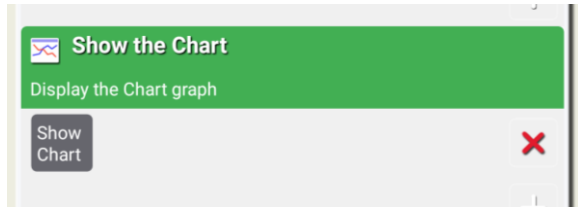


Executed when showing a list of entries and the user taps on an entry.

Show Form Entries	Show the Form with the selected entry. You can swipe between entries based on the entries from the list. Sorting and filters are also applied.
-------------------	--

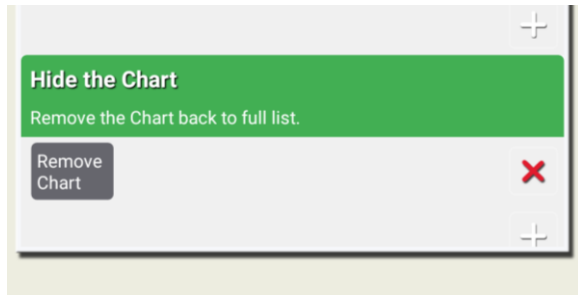


## Show the Chart



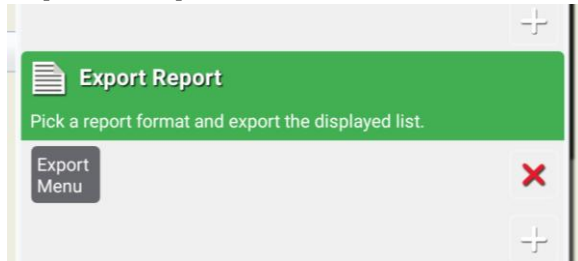
Show Chart	Show the chart window.
------------	------------------------

## Hide the Chart



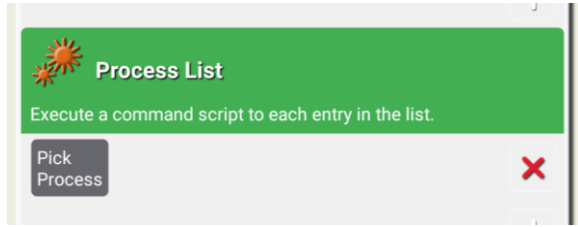
Remove Chart	Close the Chart window on the list.
--------------	-------------------------------------

## Export Report



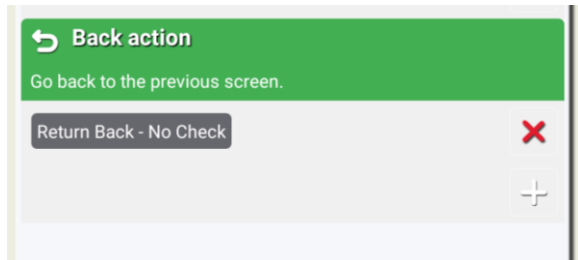
Export Menu	Show the list export choices to export the list.
-------------	--

## Process List



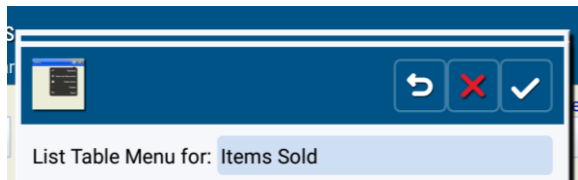
Pick Process	Show the list of Processes to pick and run.
--------------	---

## Back action



Return Back - No Check	The action when the user presses back. Return back to the last screen.
------------------------	---

## Embedded Pivot Table Menu

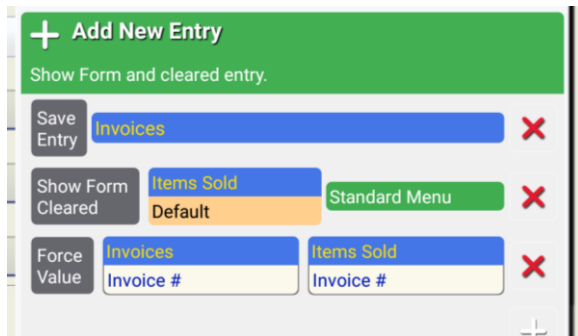


The pivot table allows you to control what happens when you tap on an entry or tap on the + key to add a new entry.

The pivot table knows a little more about what fields are used to look up the pivot table values, so the menu commands add some field movement to help out.

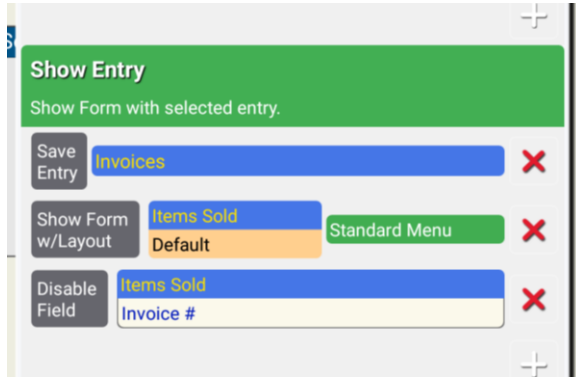
## Add New entry

Executed when the user taps on the + at the top right of the table.



Save Entry	Save any changes to this form to make sure all of the values are saved.
Show Form Cleared	Show the form that the pivot table is displaying. Cleared ready for a new entry.
Force Value	Move the Unique Key to the pivot table form, so that the saved record will be already linked to the form with the embedded pivot table. This will keep the new entry linked to this entry. This is not a requirement, but a convenience. You could remove this to allow any value.

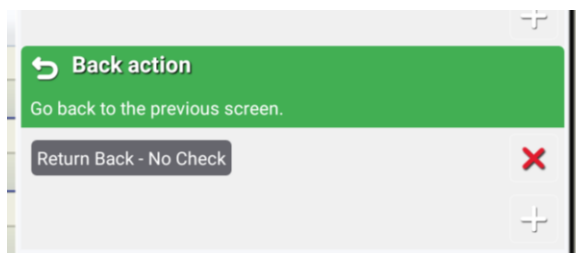
## Show Entry



Executed when the user taps on a value in the table.

Save Entry	Save any changes to this form to make sure all of the values are saved.
Show Form Entries	Show the Form with the selected entry. You can swipe between entries based on the entries from the pivot table. Sorting and filters from the pivot table list are also applied.
Disable Field	Disable the Unique Key field so that the saved record key link cannot be changed. This will keep the changed entry linked to this entry. This is not a requirement, but a convenience. You could remove this to allow any value.

## Back action



Return Back - No Check	The action when the user presses back. Return back to the last screen.
------------------------	--

## Button Commands

List Actions		Button Cmd	List	Table	List Proc
Show Form as List Entries	<p>Show the form with the selected list entry. Allow left and right swipe to switch between other entries in the list.</p> <p>Selects the <b>Default</b> Layout.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the list.</p>		✓	✓	
Show Form as List Entries w/Layout	<p>Same as <b>Show Form as List Entries</b> and allows you to pick which layout is used.</p>		✓	✓	
Show Form as single Entry	<p>Show the form with the list entry as a single entry. Do not allow left and right swipe to show other entries.</p> <p>Selects the <b>Default</b> Layout.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the list.</p>		✓	✓	
Show Form entry w/Layout	<p>Same as <b>Show Form as single Entry</b> and allows you to pick which layout is used.</p>		✓	✓	
Show Cleared Form for new entry	<p>Show the form, cleared and ready for a new entry.</p> <p>Selects the <b>Default</b> Layout.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the list.</p>		✓	✓	
Show Clear Form w/Layout	<p>Same as <b>Show Cleared Form for new Entry</b> and allows you to pick which layout is used.</p>		✓	✓	
Backup Single Form	<p>Backup the current list form as a single form.</p>		✓		

Form Actions		Button Cmd	List	Table	List Proc
Go to Form	<p>Jump to the form and show it using the <b>Default</b> layout.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the list.</p>	✓			
Go to Form Layout	<p>Jump to the form and show it using the selected layout.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the list.</p>	✓			
Show Form Layout	<p>Switch the layout on this form, but do not jump to it.</p>	✓			
Go to List	<p>Show a forms' Saved List as a list view.</p> <p>Select the list layout. Pick Default if no list layouts are saved, or create and save a saved list.</p> <p>Uses standard menu or setup a custom Menu.</p> <p>Use Return Back to return back to the form.</p>	✓			
Search Form	<p>Search and load an Entry in a Form. If not found the command processing stops.</p>	✓			✓
Put & Get Values		Button Cmd	List	Table	List Proc
Force Value	<p>Put a field value into another field, then Disable the field so it cannot be changed. Values can be within a form or from one form to another. If this command is used in conjunction with a <b>Go To Form</b>, add these commands after the <b>Go To Form</b>, otherwise the <b>Go To Form</b> will clear these values when the record loads.</p>	✓	✓	✓	✓
Put Value	<p>Put a field value into another field. If this command is used in conjunction with a <b>Go To Form</b>, add these commands after the <b>Go To Form</b>, otherwise the <b>Go To Form</b> will clear these values when the record loads.</p>	✓	✓	✓	✓

Put & Get Values		Button Cmd	List	Table	List Proc
Put Variable	Put a Group Variable into a field.	✓	✓	✓	✓
Force Variable	Put a Group Variable into a field and then Disable the field so it cannot be changed	✓	✓	✓	✓
Set Variable	Set a Group Variable with the value in field.	✓	✓	✓	✓
Clear Variable	Set a Group Variable to blank or it's default value if defined.	✓	✓	✓	✓
Notify Changed	<p>If you change values in a remote form with button commands, you may want to fields in this form to reload their values, such as an embedded table or remote fields.</p> <p>The field you pick can be the field you want to update or a key field that causes other connection link fields to load.</p> <p>If you have a table or remote field that uses a key field to look up values, you can notify the key field that it has changed and all of the values that depend on that key field will reload their values.</p>	✓	✓	✓	✓
Enter Math Equation	Use to setup a Number math or Text Math equation, to be applied to a field when when executed, just like assigning a math equation to a field. This equation is only executed when this command is executed. The Field will be disabled after the math equation executes. You can add an <b>Enable Field</b> command to afterwards to make this field available for entry again.	✓	✓	✓	✓

Put & Get Values		Button Cmd	List	Table	List Proc
Uri Intent	<p>Execute the value in the field as an Android Intent.</p> <p>Pick the Intent Action</p> <p>Use URL encoding to pass Intent Extra Values.</p> <p>To send an SMS use <b>ACTION_SENDTO</b></p> <p>Set your math equation to  "smsto:"+[<b>PHONE</b>]+"?sms_body="+encode(<b>[BODY]</b>)  ?sms_body="+encode(<b>[BODY]</b>) creates the extra "sms_body" to the encoded value.</p> <p><b>[PHONE]</b> is your phone number field  <b>[BODY]</b> is the value to send  encode(<b>[BODY]</b>) Math command encodes the value to use as a Uri  Use '&amp;' to add more extra values.</p>	✓	✓	✓	✓
Entry Actions		Button Cmd	List	Table	List Proc
Save Entry	<p>Save or Replace the entry.</p> <ul style="list-style-type: none"> <li>• Check for valid values. Stop and do not execute any more command if not valid.</li> <li>• Search and get remote form entries.</li> <li>• Calculate values.</li> <li>• Put remote values.</li> <li>• Save entry.</li> </ul>	✓	✓	✓	✓
Replace Entry if saved	<p>Replace the entry only if it has been saved and changed.</p> <p>If the entry has changed from last time it was saved, then</p> <ul style="list-style-type: none"> <li>• Check for valid values. . Stop and do not execute any more command if not valid.</li> <li>• Search and get remote form entries.</li> <li>• Calculate values.</li> <li>• Put remote values.</li> <li>• Replace entry.</li> </ul>	✓	✓	✓	✓



Entry Actions		Button Cmd	List	Table	List Proc
Reprocess Remotes	<p>Use to Replay the Save, but not create a new entry.</p> <p>First you should set all of your remote values to zero, or to the initial value you had before you saved the values in the list.</p> <p>Cause all of the Connection Links in the entry to process the Searches, Gets, Puts, Add to and Subtract from commands, as if this is a <b>new</b> entry</p> <p>Use this to replay the save of this form's entries to recreate the original transactions.</p> <p>The entry being processed is not changed and replaced. Only Remote forms are changed.</p>				✓
Save As New	<p>Save as a new entry.</p> <ul style="list-style-type: none"> <li>• Check for valid values. . Stop and do not execute any more command if not valid.</li> <li>• Search and get remote form entries.</li> <li>• Calculate values.</li> <li>• Put remote values.</li> <li>• Save entry.</li> </ul>	✓	✓	✓	✓
Delete Entry	<p>Delete the entry and show next record if available or return back if none.</p> <p>Delete the entry. Deleted forms with Active Connections will reverse their <b>Add To</b> and <b>Subtract From</b>.</p>	✓	✓	✓	✓
Delete Entry without Updates	<p>Delete the entry and show next record if available or return back if none.</p> <p>Delete the entry.</p> <p>Active Connections with <b>Add To</b> and <b>Subtract From</b> will not be reversed</p>	✓	✓	✓	✓
Clear Form	Clear the form and ready for a new entry.	✓	✓	✓	✓

Reports & Export		Button Cmd	List	Table	List Proc
Pick PDF Form	Pick a PDF Layout and print the entry using that layout.	✓			
Print PDF Layout	Print an entry using a specified PDF Layout	✓			
Export Text Report	Generate a report in text format. Preset the report encoding type. The user is shown a dialog after the report is exported.		✓		
Export CSV Report	Generate a report in CSV format. Preset the report encoding type. The user is shown a dialog after the report is exported.		✓		
Export List Choices	Show the popup dialog of Export List Choices, to allow the user to pick the report format.		✓		
Pick Process List	Show the list of Processes to Pick, Add or Edit.		✓		

Return Back		Button Cmd	List	Table	List Proc
Return Back - No Check	Leave the entry like pressing back, but do not check if it has changed and prompt Save or Ignore.	✓	✓	✓	✓
Return Back - Check Changed	Leave the entry like pressing back and check if it has changed. If it has changed, prompt Save or Ignore. If changed, show the SAVE or IGNORE alert.	✓	✓	✓	
Entry Fields		Button Cmd	List	Table	List Proc
Disable Field	Keep the user from changing this value.	✓	✓	✓	
Enable Field	Allow the user to be able to change this value.	✓	✓	✓	
Hide Field	Make a field disappear from the entry form. This has no other effect on the field, it's math and connections, other than making it invisible.	✓	✓	✓	
Show Field	Make a hidden Field on an entry form be shown again.	✓	✓	✓	
Set Focus	Move the Focus to the selected Entry Field.	✓	✓	✓	
Chart Actions		Button Cmd	List	Table	List Proc
Show the Chart	Change the list view to chart mode to show the graphic chart.		✓		
Remove the Chart	Remove the chart from the list view.		✓		

## Setting up Custom Menus

The custom menu choice let's you see and modify Standard menu actions.

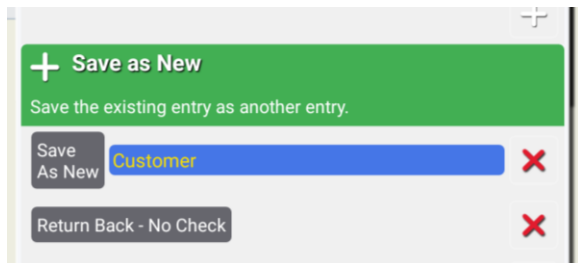
Don't worry about losing the Standard menu commands. If you don't like your commands, press the Red Delete X and the commands will be reset to the default values.

The Standard menu commands will always work, but there are times when you want to control the flow and limit what a user can press.

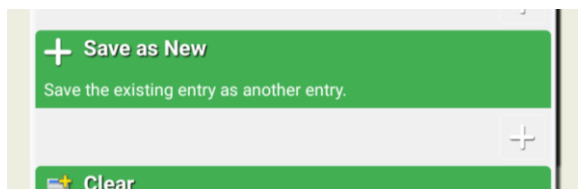
The custom menus let you modify, add, remove the Menu commands and control what actions occur when selected.

### Removing a Menu Choice.

If you do not want a menu choice available, just delete all of the commands for that menu choice. This includes the back key. Press the Red X next to the command to delete it.



Menu choices with no commands will not show.



## Modifying a Menu Choice

To modify a menu's command, either change the command or delete and add another. Tapping on the command parameters or the command itself will let you change it.

### Some custom Menu tricks.

This command set is used in the Tea Shop application.

Command	Source	Target	Action
Show Form Cleared	Received Item	Standard Menu	X
Force Value	Purchase Item	Received Item	X
Force Value	PO Number	PO Number	X
Force Value	Item SKU	Item SKU	X
Force Value	Purchased	Qty Purchased	X
Force Value	Unit Cost	Purchase Cost	X
Disable Field	Received Item	Item Name	X

When viewing the Purchase Orders, when a user picks one (Show Entry), the system will load that entry into the Purchase Order Form.

From there, we show the Received Items Form.

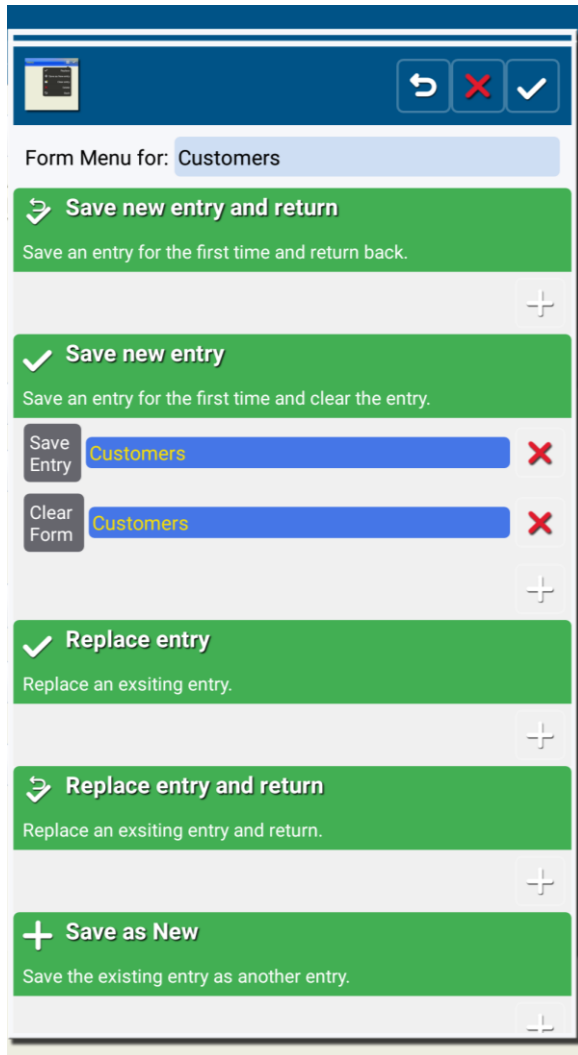
We Force the Po Number into the Received Items PO Number.

We Force the Distributor # into the Received Items Distributor #.

We disable the Distributor Field so the user can't change it.

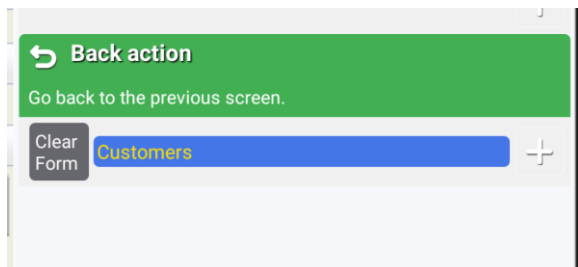
Then we save the entry so we can maintain these values. This gets deleted when we are done. That is done with the Back action as see here.

## Removing all menus to force a user.



## Disabling the Android Back button.

If you remove the Back Action command, the button will not show, but the Android back key will still work. If you want to disable the Android back key, to lock the user in the screen, you need to define a command on the Back Action. Using Clear Form works well.



**Note:** You will not be able to exit Sailforms if you implement this. This can be useful where you are creating a Log In page for users and you do not want them pressing the Android Back button to exit the screen.

## Remote Form Entries and Searching

SailformsPro was built to make hooking up relational forms easy.

More complex relational demands has forced more complex relational setup.

### Simple Connections

When you add a "Search-List-Get" field to a form, you are setting up a relation to a remote form.

Once that remote form entry is selected with your "Search-List-Get" field, the other fields in that form can be accessed with the other Active Connection commands.

#### Active Connections

Get Remote Value

Put Remote Value

Add To

Subtract From

#### Remote Controls

Remote Field

All of the above Active Connections work with the same entry value found with Search-List-Get. There is only 1 remote entry and it must be a different form.

The menu and button commands also can only apply to that single remote entry.

Using Put Value, Force Value, hide field, disable field and executing a math equation only apply to that single remote entry for that form.

### Recursive Lookup

When recursive lookup was added to Sailforms, we added a form cache so a form could be connected to other forms and other instances of itself.

We also added commands to be able to search and load values using key lookup.

## **New Recursive Lookup Functions**

### Active connection

Look up

### Remote Control

Remote Search

Total Remote

Remote Count

Maximum Remote

Minimum Remote

These commands and controls work with their own instances of the remote forms. They are accessing the database directly to load their values, so that there is no instance of the remote form's entry to apply transactions to. You can not use get/Put/Add or Subtract.

Since there is a need to be able to search and load a record and update a field with a math command, we added Search Form to the button commands.



### **Search Form**

Search form let's you lookup any remote entry, put values, do math commands and save the remote entry using menu commands or button commands.



## How Sailforms picks layouts

When Sailforms picks a screen layout size for a device, it goes through the following tests to determine the correct choice.

1) For the current rotation, Landscape  or Portrait  , if there is a layout that matches the screen size, then it is used.

There are three values that make up the screen size.

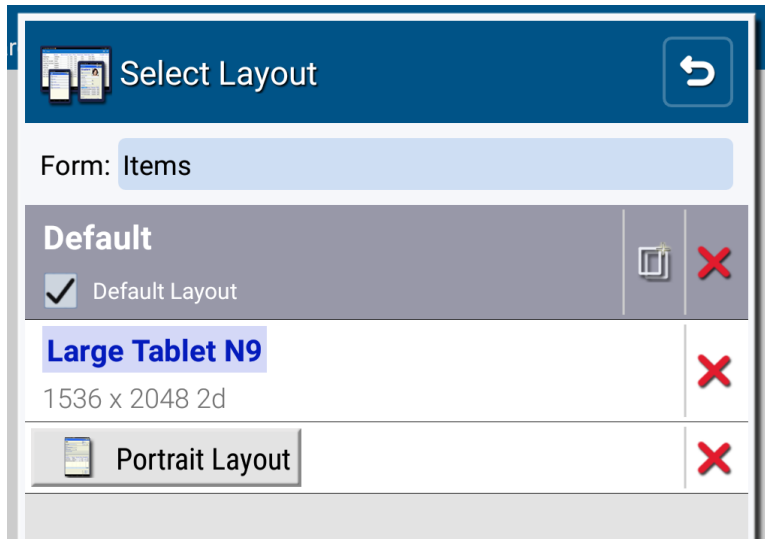
- Width
- Height
- Density

2) If the screen is bigger, the biggest layout is picked. It is centered horizontally.

3) If there is a Portrait layout that will fit in the device in Landscape, the screen will flip to Landscape and the Portrait layout will be used.


4) If there is no layout that will fit, then the layout is horizontally scaled down to fit.

## Using Alternative Layouts



Every form requires a default layout.

Anytime **Sailforms'** needs a layout, if there is no layout specified, then the Default layout is used.

Press the copy button  to create an Alternative Layout.

Enter a new layout name and save the new Layout.

All of the screen layouts will be copied into the new layout.  
You can edit or modify each of those separately.

When using Button commands, when you move to another Form you can select an Alternative Layout, or use the Default.

You can also change the layout on your current form, using a button with the Show Form Layout button command.