

Customizing SideKick 365 CRM



Version 1.0.0

8/30/2018

Created and Built in California by SkyLite Systems

Copyright 2018

Preliminary Software

Let's get started. We need to install some components on your PC to make the add-ins run. You need to install the SharePoint 2013 runtime library and .NET 4.7.2. This section walks through how to install these two components.

.NET 4.7.2

Please navigate to the link below and install .NET 4.7.2

<https://www.microsoft.com/net/download/dotnet-framework-runtime>

Visual Studio 2010 Tools for Office Runtime

Please navigate to the link below and install Visual Studio 2010 Tools for Office Runtime

<https://www.microsoft.com/en-us/download/details.aspx?id=48217>

SharePoint 2013 Client Runtime SDK

Now, go to <http://www.microsoft.com/en-US/download/details.aspx?id=35585> and download and run the SharePoint Client runtime library. Download and install the file "sharepointclientcomponents_15-4711-1001_x86_en-us.msi" if you are running a 32 bit version of Windows. Download and install "sharepointclientcomponents_15-4711-1001_x64_en-us.msi" if you are running a 64 bit version of Windows

If you are running Windows 8 or 10 the follow the instructions here after you install .NET 4.7.2

<http://dynamicsofdynamicscrm.com/2015/09/11/quick-tipenabling-windows-identity-foundation-on-windows-10-machines/>

If you are running Windows 7 then Download Windows Identity Manager here

<https://www.microsoft.com/en-us/download/details.aspx?id=17331>

After installation completes, you will find the Microsoft.IdentityModel.dll in the GAC, and the Claims To Windows Token Service executable file, c2wtshost.exe, in the directory C:\Program Files\Windows Identity Foundation\v3.5\.

IMPORTANT - Restart your computer after you install these updates.

Building Custom Columns and Screens using SideKick365 xRM Manager

This guide will show you to use SideKick365 xRM Manager to build incredibly rich forms that will let you leverage all the features of SharePoint using HTML and JavaScript. All data entered in the forms is stored within SharePoint lists. That means you can leverage all of the great capabilities of SharePoint such as custom list views, search, document management, workflows, and permission management with the forms you create. You don't need any additional servers or links to external systems to build forms with SideKick365 xRM Manager and run them in SideKick 365 CRM.

This guide includes straight-forward descriptions, instruction and detailed examples that you can study and learn from. We expect you understand the basics of HTML and JavaScript so you can follow along when we reference certain terminology or when you study the examples in the Appendix. There are countless references to HTML and JavaScript on the web that you can reference as needed if you encounter new terms or concepts.

We think SideKick 365 CRM is a breakthrough productivity tool and know you will find new ways to use it in ways we never imagined.

Please watch video 2 at the link below and then read through this manual

<https://skylitesystems.com/sidekick-365-crm-videos>

Now we are ready to get started....

Launch SideKick365 Manager

Start SideKick365 Manager and supply the url of the home page of your instance of SideKick 365 CRM.

TIP – You only need the portion of the url up to “/Pages”. For example

<https://skylite-4bf326209d0501.sharepoint.com/sites/crmdemo/SideKick365CRM/Pages/Home.aspx?SPHostUrl=https%3A%2F%2Fskylite%2Esharepoint%2Ecom%2Fsites%2Fcrmdemo&SPLanguage=en%2DUS&SPClientTag=0&SPProductNumber=16%2E0%2E8022%2E1214&SPAppWebUrl=https%3A%2F%2Fskylite%2D4bf326209d0501%2Esharepoint%2Ecom%2Fsites%2Fcrmdemo%2FSideKick365CRM>

Should be shortened to

<https://skylite-4bf326209d0501.sharepoint.com/sites/crmdemo/SideKick365CRM>

The first time you open SideKick365 xRM Manager, you will see a dialog asking you to supply the URL of the home page of SideKick 365 CRM. You will also need to provide a user name and password as shown in Figure 1 below.

TIP - The user MUST be a member of the owners group in the site that holds SideKick 365 CRM

Figure 1 – Log-in Screen

Tip – Selecting the “Keep me signed in” checkbox will cache these settings so you don’t need to supply them every time you launch SideKick365 xRM Manager.

Changing Names of the Modules

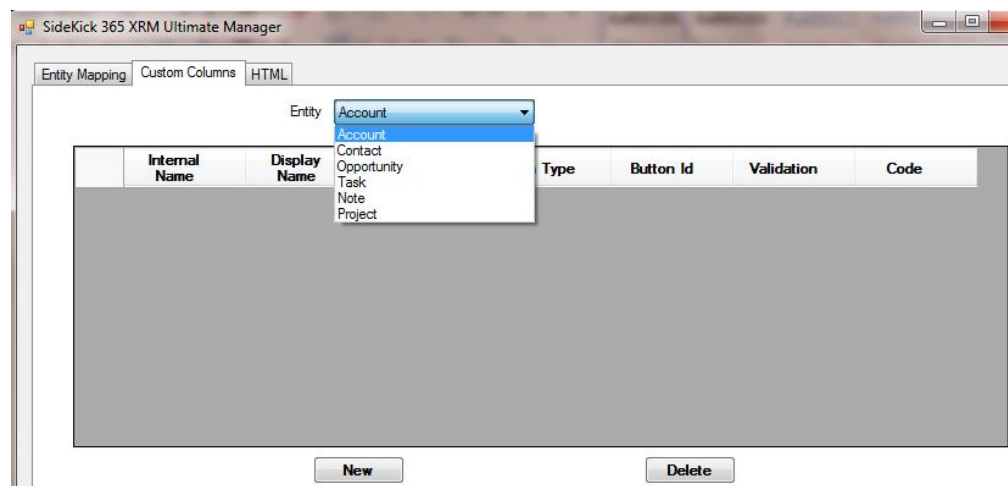
You can change the names that are displayed for each module by using the screen shown below. Make sure to hit the update button when you want these changes to apply to SideKick 365 CRM.

	Standard Name	Custom Name	List Title
Account	Account		
Contact	Contact		
Opportunity	Opportunity		
Task	Task		
Note	Note		
Project	Project		

Update

Adding Custom Columns

Use the screen below to add custom columns to any of the modules in SideKick 365 CRM. Make sure to use the Dropdown and select the proper entity (as shown)



When you create a field in SideKick 365 CRM Manager, a field is also created in SharePoint to store the data a user enters within the forms you create. It is essential that you specify a few things when you add a field so data can be stored properly in SharePoint, and the fields you define can be properly to controls within your HTML form. These items include:

Internal Name – this is a unique name used to help identify the field behind the scenes in SideKick365 pages. It is also used when you want to bind the field to a screen element using the NAME attribute in SELECT controls in HTML, in Radio Buttons, and within JavaScript and JQuery.

IMPORTANT – The Internal Name and the Screen Element Name must be the same for the following two data types used in SideKick365 Forms

- Choice
- Lookup
- Radio Button

Best Practice – We suggest you start the Internal Name with “Cstm” such as “CstmMarketing” or “CstmBanking” as shown in figure 4 below. This makes it easy to identify Internal Name and not to confuse them with the Screen Control ID (id) tag. In fact, you can name these fields anyway you wish, but we offer this guidance as a best practice.

Best Practice – You can define the values in a choice list within the list settings in SharePoint! When you define the field as a “choice” field and include the NAME and ID tags in the HTML form (they should both be the same since this is a <select> type control), the values associated with the choice list in SharePoint will be displayed within the choice list in your HTML form. Consult the Appendix for a detailed example.

Display Name – This value is used to identify the field within a SharePoint list. The value you enter here will be displayed within SharePoint as the default column name and the name you will see when you work with list setting and custom views for this list within SharePoint.

Screen Element ID – This column holds the id that is used to bind the screen element in HTML with the data in SharePoint. This is used by the ID tag and is required in all field types used within your HTML forms.

Button ID – This is used by the People Picker control and is used to reset the name on the HTML form. You must assign a button name when you create a People Picker. An example of how to use the People Picker in your HTML forms is included in the Appendix.

Best Practice – We suggest you start the Button ID with “Btn” such as “BtnManager” or “BtnSupervisor”. This is not required but we offer this guidance as a best practice.

Validation Checkbox – This button is used when you want to execute JavaScript that will run when data is entered into the field. The JavaScript should be entered into box entitled Code within the SideKick365 Manager.

Best Practice – You can also execute JavaScript or JQuery inline within your HTML or you can execute with a field by using the Code and Validation buttons shown on the sample form below

Data Types Support – The following datatypes are supported within forms you create using SideKick365 Manager

- Text
- Area
- Choice
- Number
- Currency
- Date
- People Picker
- Radio Buttons
- Picture
- Hyperlink

Adding Custom Columns

Use the screen below to add custom columns to any of the modules in SideKick 365 CRM. Make sure to use the dropdown and select the proper entity (as shown below)

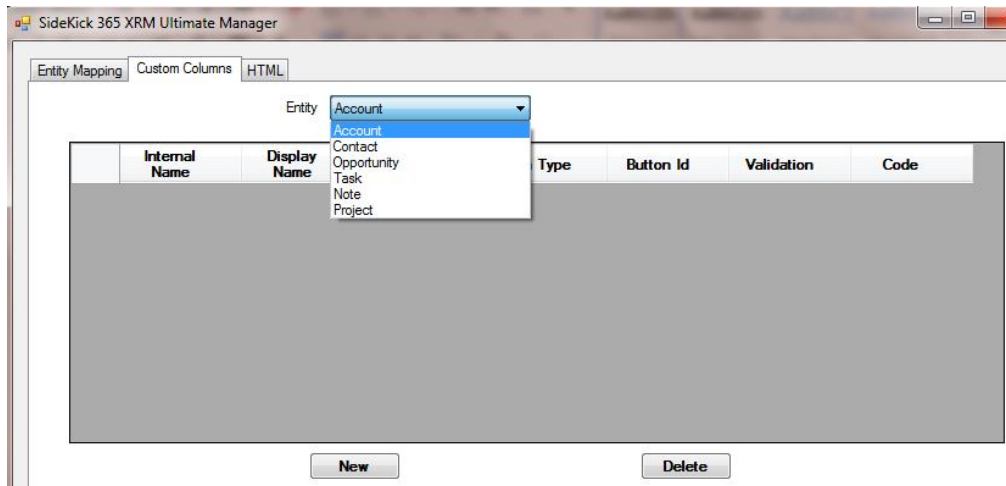


Figure 3 – Selecting an Entity to add a Custom Column

When you create a field in SideKick 365 CRM Manager, a field is also created in SharePoint to store the data a user enters within the HTML forms you create. It is essential that you specify a few things when you add a field so data can be stored properly in SharePoint, and the fields you define can be properly to controls within your HTML form.

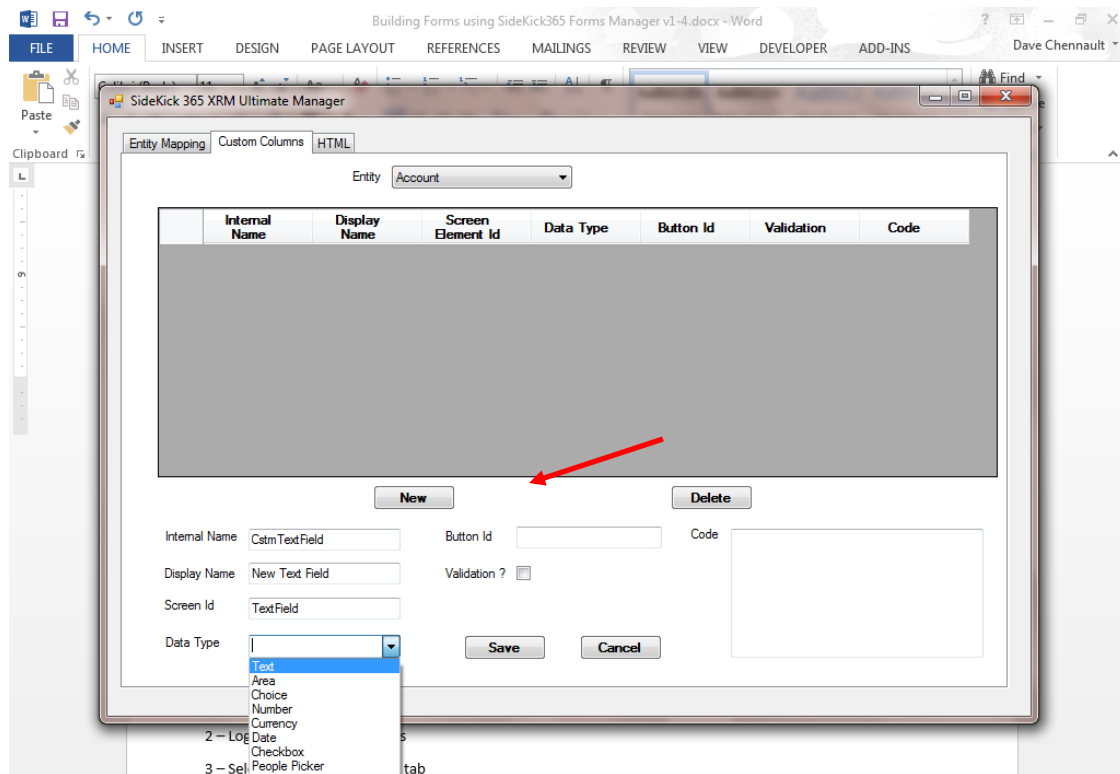


Figure 4 – Adding a Custom Column

The elements in the New Field definition screen include:

Internal Name – this is a unique name used to help identify the field behind the scenes in SideKick365 pages. It is also used when you want to bind the field to a screen element using the NAME attribute in SELECT controls in HTML, in Radio Buttons, and within JavaScript and JQuery.

IMPORTANT – The Internal Name and the Screen Element Name must be the same for the following two data types used in SideKick365 xRM

- Choice
- Lookup
- Radio Button

Best Practice – We suggest you start the Internal Name with “Cstm” such as “CstmMarketing” or “CstmBanking” as shown in figure 4 below. This makes it easy to identify Internal Name and not to confuse them with the Screen Control ID (id) tag. In fact, you can name these fields anyway you wish, but we offer this guidance as a best practice.

Best Practice – You can define the values in a choice list within the list settings in SharePoint! When you define the field as a “choice” field and include the NAME and ID tags in the HTML form (they should both be the same since this is a <select> type control), the values associated with the choice list in SharePoint will be displayed within the choice list in your HTML form. Consult the Appendix for a detailed example.

Display Name – This value is used to identify the field within a SharePoint list. The value you enter here will be displayed within SharePoint as the default column name and the name you will see when you work with list setting and custom views for this list within SharePoint.

Screen Element ID – This column holds the id that is used to bind the screen element in HTML with the data in SharePoint. This is used by the ID tag and is required in all field types used within your HTML forms.

Button ID – This is used by the People Picker control and is used to reset the name on the HTML form. You must assign a button name when you create a People Picker. An example of how to use the People Picker in your HTML forms is included in the Appendix.

Best Practice – We suggest you start the Button ID with “Btn” such as “BtnManager” or “BtnSupervisor”. This is not required but we offer this guidance as a best practice.

Validation Checkbox – This button is used when you want to execute JavaScript that will run when data is entered into the field. The JavaScript should be entered into box entitled Code within the SideKick365 xRM Manager.

Best Practice – You can also execute JavaScript or JQuery inline within your HTML or you can execute with a field by using the Code and Validation buttons shown on the sample form below

Data Types Support – The following datatypes are supported within forms you create using SideKick365 xRM Manager

- Text
- Area
- Choice
- Number
- Currency
- Date
- People Picker
- Radio Buttons
- Picture
- Hyperlink

Creating a New Field in SideKick 365 CRM

It is easy to add fields in SideKick365 xRM Manager. This section will explain the process and all of the options you have available to add all the fields you need to build rich and powerful forms. Let’s take a closer look.

Follow the steps outlined below to add a field using SideKick365 xRM Manager.

1 – Launch SideKick365 xRM Manager

2 – Put in URL, User Name, and Password. User should be a part of the Member or Owner group within the SharePoint site that holds SideKick365 xRM.

3 – Add the entity (lists) that you need for the form. Remember that you can add a lookup entity (list) as needed.

4 – Select the Custom Columns Tab to start adding fields for your form.

Tip – Build can only customize one form per module.

5 – Select the “New” button and using the form shown below, start entering in information about the new field. Make sure to fill out all the info needed including the Name, ID, Field Type and any JavaScript you want to add for validation, or a button ID if you are adding a people picker control.

6 - Continue adding fields as needed and hit the Save Button when you are ready to put these fields into action and make them available for your Form.

Best Practice – It is a good idea to enter a few fields and then hit the save button. Don’t put in everything and hit the Save button. Better to add a few at a time.

Binding a HTML Form to a list in SideKick365 xRM

SideKick365 xRM Manager lets you design rich forms in HTML and JavaScript and then bind these forms to a module (SharePoint list) in SideKick365 xRM. You can use your favorite HTML Design tool like Dreamweaver, Microsoft Expressions, Visual Studio, or even a text editor like TextPad to create your HTML layouts and JavaScript files.

Tip – SideKick365 Form also support JQuery

Best Practice– The appendix includes a sample HTML form, and JavaScript to build powerful forms and to serve as a reference when you start building your own forms.

Load HTML Button

Binding a form to a list is easy in SideKick365 xRM Manager. Just follow the steps below

1 – Modify the HTML form for a module desired by adding new HTML and JavaScript (jQuery is supported) using your favorite HTML editor or design tool.

2 – Log into SideKick365 xRM Manger and select the HTML tab

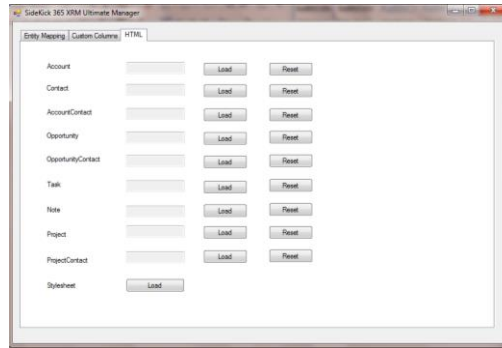
3 – Select the Module you want to bind the form to by selecting the HTML tab

4 – Select the Load Button as shown in the figure 4 below

5 – Browse to the form you want and hit OK

6– Hit Reset if you want to revert back to the Original form

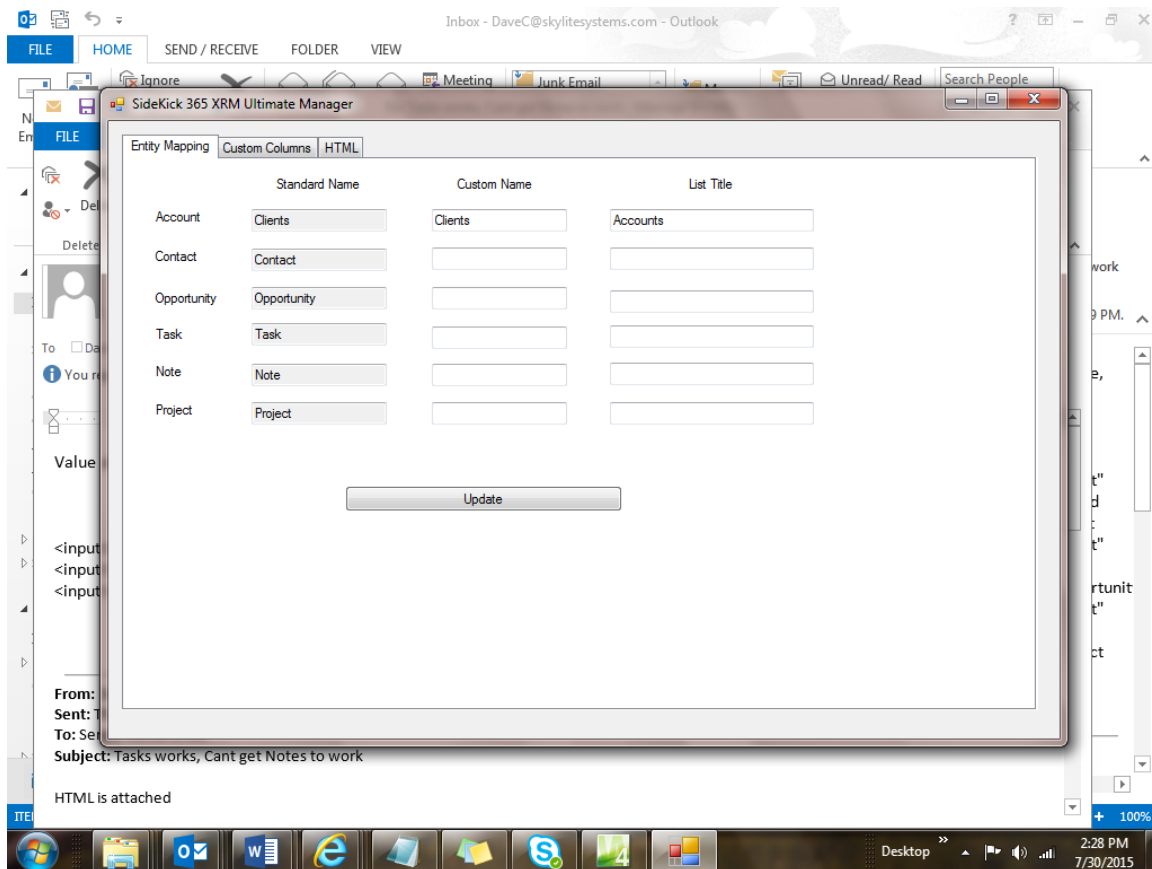
IMPORTANT – DO NOT DELETE ANY FIELDS THAT SHIP WITH SideKick 365 CRM. CONTACT SKYLITE SYSTEMS IF YOU NEED TO DELETE SOME FIELDS.



Binding an HTML Layout to a Module

Additional Changes Required if you Customize the Name of the Account, Opportunity, or Project Modules

If you change the Account, Opportunity, or Projects List names then you need to also update the HTML in Tasks and Notes screens so the new names you defined can be used to bind the Note or Task to an Account, Opportunity, or Project. The remainder of this section will show you what needs to be changed to keep Notes and Tasks “child” records bound to the “parent “ Account, Opportunity, or Project .



Appendix

The appendix contains samples that demonstrate how to work with each data type in HTML so the data entered in the HTML form is bound into the SharePoint list. Samples are generally written using JQuery.

Working with Data Types in HTML within your Forms

This section will show you how to use each datatype within your HTML forms

Tip – Most data types only require the id attribute to be defined, however, the name attribute should also be defined if you want to work with the name attribute in JavaScript or JQuery.

Definitions

When working with HTML, the following conventions are used:

HTML Attribute Convention

id

name

SideKick365 xRM Manager Naming

Screen Element ID

Internal Name

Datatypes

The remaining section of this guidewill show you how to work with each custom column data type supported in SideKick365 xRM Manager.

Tip - The HTML File and Custom Column .CSV file is also included for your reference. You can import the Custom Column Definition and attach this HTML file to see SideKick 365 CRM and SideKick365 xRM Manager in action!

Text

```
<!-- Here is an example of how to work with Text fields  
You must set the id element to equal the screen element ID you defined  
in SideKick365 Forms Manager... that is the only requirement!  
-->
```

```
<tr>  
  <td style="width: 191px">Text</td>  
  <td colspan="3">  
    <input id="TextFieldExample" type="text" style="width: 887px">  
  </td>  
</tr>
```

Area

```
<!-- Here is an example of how to work with an Area field  
Use Area fields when you have a lot of text and you want a scrolling text box  
You must include ID field with the text field... that is the only requirement!  
-->  
<tr>  
  <td style="width: 191px" valign="top">Area</td>  
  <td colspan="3">  
    <textarea id="AreaExample" rows="5" style="width: 885px">Please enter some text here :</textarea>  
  </td>  
</tr>
```

Choice

<!-- Here is an example of how to work with an a choice list (dropdown)
HTML uses the Select tag to handle the Choice lists
This is a very interesting control! It pulls the choices that you define in SharePoint right into the form :) You must include ID field and the name fields when you use the Select tag ...
The Screen Element ID and the Internal Name that you define in SideKick365 Forms Manager must be the exact same or you wont see the choices that you defined in SharePoint within the form.

You can Manage the choices presented in the form by following the steps below:

- 1 - go to list of forms
 - 2 - Give 1 click on any line so the checkmark appears on he row and the tabs appear at the top of the list
 - 3 - Select the List tab
 - 4 - Select the Modify View option within the ribbon
 - 5 - Now move the word over the word SETTINGS (it is a big font on the list view set up page)
 - 6 - Now look for the field you defined as a choice in SideKick365 Forms Manager
 - 7 - Click on the name of that Choice field
 - 8 - Now you can manage the choices preseneted by managing the choices within this screen
- >

```
<tr>
  <td style="width: 191px">Choice</td>
  <td colspan="3">
    <select name="CstmChoiceListExample" id="CstmChoiceListExample" style="width: 885px">
      <option></option>
    </select>
  </td>
</tr>
```

Number

<!-- Here is an example of how to work with Number fields
You must set the id element to equal the screen element ID you defined in SideKick365 Forms Manager... that is the only requirement!
-->

```
<tr>
  <td style="width: 191px">Number</td>
  <td colspan="3"><input type="number" id="NumberExample"> </td>
</tr>
```

Currency

<!-- Here is an example of how to work with Currency fields
You use the type= number and set the id element to equal the screen element ID you defined in SideKick365 Forms Manager... that is the only requirement!
-->

```
<tr>
  <td style="width: 191px">Currency</td>
  <td colspan="3"> $ <input type="number" id="CurrencyExample" ></td>
</tr>
```

Date

```
<!-- Here is an example of how to work with Date fields
You use the Type = date and set the id element to equal the screen element ID you defined
in SideKick365 Forms Manager... that is the only requirement!
-->

<tr>
  <td style="width: 191px">Date</td>
  <td colspan="3"><input type="date" id="DateExample" ></td>
</tr>
```

Checkbox

```
<tr>
  <td style="width: 191px">Checkbox</td>
  <td colspan="3">
    <input type="checkbox" id="CheckboxExample" value="Checkbox 1" style="width: 29px"> Checkbox1</td>
</tr>
```

People Picker

<!-- Here is an example of how to work with People Picker Control
This is an interesting control and a lot of complexity is handled by SideKick365 Forms
This control has the following components, the field that holds the peoplepicker value,
and the Button used to reset the value in the PeoplePicker.

In the first cell <td>, set the id = to the Screen Element ID of the peoplepicker
you created in SideKick365 Forms

In the second cell <td> set the id of the button equal to the Button Id that you set up
for the peoplepicker in SideKick365 Forms Manager.

In the example below, the PeoplePicker Screen id was set to PeoplePickerExample and
the button is called BtnPeoplePickerExample. You must define the name of the Button
in SideKick365 Forms Manager when you set up a People Picker field.

```
-->

<tr>
  <td style="width: 191px">People Picker</td>
  <td style="width: 4%"><div id="PeoplePickerExample"/></td>
  <td style="width: 4%"><input type="button" value="Reset" style="display:none; width:25px"
    id="BtnPeoplePickerExample" onclick="reset_pp(this.id);" /></td>
</tr>
```

Hyperlinks

```
<!-- HyperLinks
You will need to create one field in SideKick365 Forms using Manager that is of the type Hyperlink.
In the HTML below, we have a field with the Screen Element ID = HyperLinkExample.
You can add whatever name you want to your solution but make sure to replace
the term HyperLinkExample with your field when you use the example below
Make sure to replace the word HyperLinkExample with your hyperlink Screen Element ID everywhere you
see it in the example below, but make sure to leave any other code in place.

For example, lets say you had a hyperlink column called company_website, using the
example below, replace HyperLinkExample_edit_div with company_website_edit_div,
and replace HyperLinkExample_url with company_website_url. Next replace HyperLink_description
with company_website_description. Finally replace HyperLinkExample_display with company_website_display.

We have code behind the form that will handle formatting everything for you... try it!
-->

<tr>
  <td style="width: 191px">Hyperlink</td>
  <td colspan="3">
    <div id="HyperLinkExample_edit_div" style="display:inline;">
      URL&nbsp;&nbsp;&nbsp;<input type="url" id="HyperLinkExample_url" value=""><br/>
      Description&nbsp;&nbsp;&nbsp;<input type="text" id="HyperLinkExample_description" value="" />
    </div>

    <div id="HyperLinkExample_display_div" style="display:none">
      <a id="HyperLinkExample_display" href=""></a>
    </div>
  </td>
</tr>
```

Picture

```
<!-- Pictures - You will need to create one field in SideKick365 Forms using Manager that is
of the type Picture. In the HTML below, we have a field with the Screen Element ID = PictureExample.
You can add whatever name you want to your solution but make sure to replace
the term PictureExample with your field when you use the example below
Make sure to replace the word PictureExample with your Picture Screen Element ID everywhere you
see it in the example below, but make sure to leave any other code in place.

TIP - Notice that we reference a url... you need to store teh pictue somewhere (could be a document library)
And then reference the url to that puicture

For example, lets say you had a picture column called company_picnic, using the
example below, replace PictureExample_edit_div with company_picnic_edit_div,
and replace PictureExample_url with company_picnic_url. Next replace Picture_description
with company_wp picnic_description. Finally replace PictureExample_display with company_picnic_display.

We have code behind the form that will handle formatting everything for you... try it!
-->
<tr>
  <td style="width: 191px; height: 23px;" valign="top">Picture</td>
  <td class="pm-formbody" colspan="3">
    <div id="PictureExample_edit_div" style="display:inline;">
      URL&nbsp;&nbsp;&nbsp;<input type="url" id="PictureExample_url" value="" style="width: 592px"/><br/>
      Description&nbsp;&nbsp;&nbsp;<input type="text" id="PictureExample_description" value="" style="width: 593px" />
    </div>

    <div id="PictureExample_display_div" style="display:none">
      <img id="PictureExample_display" src="" alt="" width="600" height="460"><br/>
      <a id="PictureExample_description" href=""></a>
    </div>
  </td>
</tr>
```


Radio Buttons

```
<table style="width: 75%" >
<!-- Here is an example of how to work with Radio Buttons
You must include the Name and the ID elements
The Name element is used so the radio button functions properly - only 1 button in the group is selected at a time
the Name element MUST be the same as the Internal Name defined in SideKick365 Forms Manager
The Id is used to bind the data in Sharepoint with the button
Notice that all the buttons in teh group share the same name and id
NOTE: An onload event is used to ensure that the choice 1 button is selected if this is a new form
that event is a JQuery file that is set up in SideKick365 Manager using the Custom Code tab and associating it
with the onload event for this form -->

<tr>
<td style="width: 191px" >Radio Buttons</td>
<td > <input type="radio" name="CstmRadioButtonExample" id="CstmRadioButtonExample" value ="Radio 1" onclick="$
<td > <input type="radio" name="CstmRadioButtonExample" id="CstmRadioButtonExample" value ="Radio 2" onclick="$
<td > <input type="radio" name="CstmRadioButtonExample" id="CstmRadioButtonExample" value ="Radio 3" onclick="$
</tr>
<tr></tr>
</table>
```

Here is the Raw Text for Examples Above

```
<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Untitled 1</title>

</head>

<body>

<table style="width: 100%">

<!-- This is how you can add a text field -->

    <tr>

        <td class="pm-label"><span class="pm-standardheader">Text Field Name</span></td>

        <td class="pm-formbody"><input class="pm-shortinput" type="text" id="" value="" /></td>

    </tr>

<!-- End of how you can add a text field -->

<!-- This is how you can add a number field -->

    <tr>

        <td class="pm-label"><span class="pm-standardheader">Number Field Name</span></td>

        <td class="pm-formbody"><input class="pm-shortinput" type="number" id="" value="" /></td>

    </tr>

<!-- End of how you can add a number field -->
```

<!-- This is how you can add a picklist field -->

```
<tr>

    <td class="pm-label" style="height: 26px"><span class="pm-standardheader1">Picklist Field
Name </span></td>

    <td class="pm-formbody" style="height: 26px"><select class="pm-shortinput" name=""
id=""></select></td>

</tr>

<!--End of Picklist example-->
```

<!-- This is how you can add a required text field -->

```
<tr>

    <td class="pm-label"><span class="pm-standardheader1">Region</span><span
style="color:red">&nbsp;*</span></td>

    <td class="pm-formbody" style="width: 4px"><input class="pm-shortinput" type="text" id=""
value="" /></td>

</tr>

<!--End of required text field example-->
```

<!-- Here is how to add a Radio Button -->

```
<tr>

    <td class="pm-label"><span class="pm-standardheader">Type </span></td>

    <td class="pm-formbody" colspan="1">

        <input type="radio" name="acc_Temp" id="acc_Temp" value="NA" checked />NA

        <input type="radio" name="acc_Temp" id="acc_Temp" value="Hot" />Hot

        <input type="radio" name="acc_Temp" id="acc_Temp" value="Cold" />Cold

    </td>

</tr>

<!--End of Radio Button example-->
```

<!-- Here is how to add a PeoplePicker -->

```
<tr>

    <td class="pm-label">

        <span class="pm-standardheader">Back Up Manager</span>

    </td>


    <td class="pm-formbody">

<table>
<tr>
    <td>

        <div id="acc_Backup"/>

    </td>


    <td>

        <input type="button" value="Reset" style="display:none; width:25px"
id="btnResetBackUp" onclick="reset_pp(this.id);" />

    </td>

</tr>
</table>

</td>

</tr>
```

<!-- end of PeoplePicker example -->

```

<!-- Here is how to add a Hyperlink -->
<tr>
    <td class="pm-label" >
        <span class="pm-standardheader">HyperLink</span>
    </td>

    <td class="pm-formbody" colspan="1" >
        <div id="accountlink_edit_div" style="display:inline;">
            Description&nbsp;&nbsp;&nbsp;<input class="pm-longinput" type="text"
id="accountlink_description" value="" />
            <br/>
            URL&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input
class="pm-longinput" type="url" id="accountlink_url" value="" />
            </div>

            <div id="accountlink_display_div" style="display:none">
                <a id="accountlink_display" href=""></a>
            </div>
        </td>
</tr>
<!-- End of hyperlink example -->

```

<!-- Picture Example -->

<tr>

<td class="pm-label">

Image

</td>

<td class="pm-formbody" colspan="3">

<div id="accountphoto_edit_div" style="display:inline;">

URL <input class="pm-longinput" type="url"
id="accountphoto_url" value=""/>

Description <input class="pm-longinput" type="text"
id="accountphoto_description" value=""/> />

</div>

<div id="accountphoto_display_div" style="display:none">

</div>

</td>

</tr>

<!-- End of Picture Example -->

<!-- this is how to add an area field-->

```
<tr>
    <td class="pm-label" style="vertical-align:top"><span class="pm-standardheader">Long
Notes Header</span></td>
    <td class="pm-formbody" colspan="3" rowspan="3" style="vertical-align:top"
class="pm-formbody"><textarea rows="7" cols="125" class="pm-textarea" id=""></textarea></td>
</tr>
<!-- end of area example-->
```

```
</table>
```

```
</body>
```

```
</html>
```

Reserved Fields

SideKick365 xRM has a few reserved fields. These include

Mode

- New
- Edit
- Display

Name – this appears under the Title column in list views of the form