3
Introducing Custom Fields

As you've worked your way through the first two chapters of this book, you have learned how to:

- Change the title of each of the module tabs that make up the SugarCRM front end.
- Change the terminology used in each of the module tabs, so that SugarCRM uses the same terms as your organization.
- Give SugarCRM a look and feel in keeping with your company's branding.
- Create your own module tabs and dashlets, either to add your own functionality, or to incorporate any existing web-based applications already used in your organization.

So, now you can provide a SugarCRM implementation that won't be completely alien to your intended users—hopefully they'll be able to use the application with minimum training.

Of course, there's more to it than just making SugarCRM look the way that you want. Let us imagine Korora Blue sat at her desk. The first thing that she does, every morning, is to evaluate all of the new preliminary investigations. When she does this then she decides if any surveillance needs to be carried out. As it stands, there is nowhere to store this in the SugarCRM application. So, obviously, we need to provide some extra fields for Korora so that she can do her job.

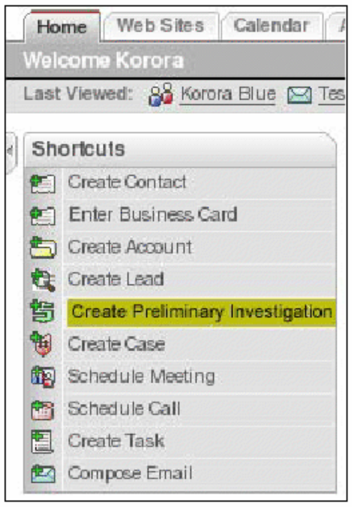
And that's the aim of this chapter—to show you how to add your own custom fields to SugarCRM. By the end of the chapter your implementation won't just look different to the standard, out-of-the-box SugarCRM application, it will actually start to behave differently.

Adding a Custom Field

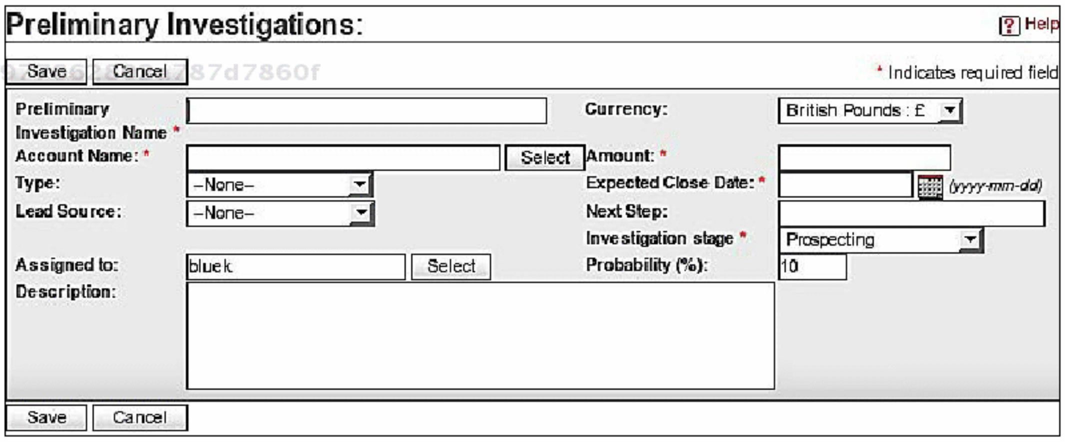
Before we jump in and add a new field let's have a look at what it is that Korora is trying to achieve.

The Standard Module Tab

When Korora decides that a new preliminary investigation is required she can do so by using the shortcuts on her **Home** tab:



Having clicked on **Create Preliminary Investigation** she just needs to fill in the appropriate details:



Type:	–None–	Currency:	British Pounds : £
Lead Source:	–None–	Investigation stage *	Prospecting
Assigned to:	bluek	Probability (%):	10

So, is there a show stopper here? Well, in Korora's case, yes. She needs to be able to record whether or not any surveillance is required, but that's not possible with the standard form. What she actually needs is an extra drop-down field with a 'Yes/No' option.

The General Process for Creating a Custom Dropdown

Having decided that Korora needs a dropdown adding to **Preliminary Investigations** we need to go through three stages in order to add the custom field:

1. Create the options for the drop-down box
2. Create the custom field itself and link the options to it
3. Place the drop-down box on a module tab

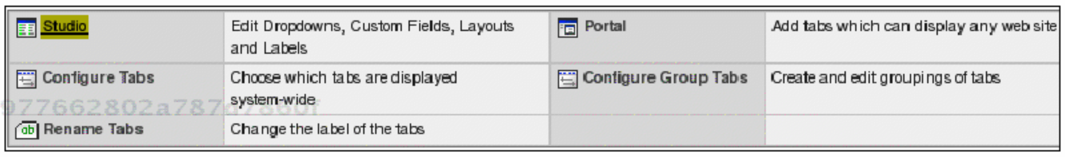
To start with there are two ways to create your own drop-down box:

- By using Studio
- Manually

We'll start by using Studio.

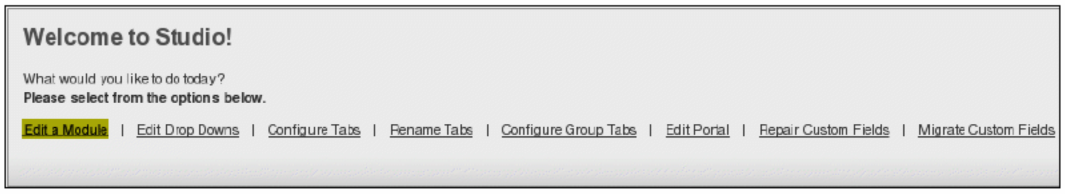
Using Studio to Create a Drop-down Box

As you would expect you will need to log on to SugarCRM as an administrator, and then go to the **Admin** screen where you'll find the link to the **Studio**:



Studio	Edit Dropdowns, Custom Fields, Layouts and Labels	Portal	Add tabs which can display any web site
Configure Tabs	Choose which tabs are displayed system-wide	Configure Group Tabs	Create and edit groupings of tabs
Rename Tabs	Change the label of the tabs		

Once you're in the Studio, you need to find the **Edit a Module** link:

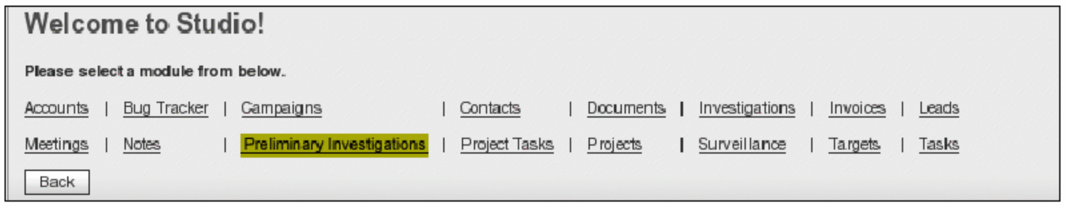


Welcome to Studio!

What would you like to do today?
Please select from the options below.

Edit a Module | Edit Drop Downs | Configure Tabs | Rename Tabs | Configure Group Tabs | Edit Portal | Repair Custom Fields | Migrate Custom Fields

And then choose the module that you want to edit. At the moment we're interested in **Preliminary Investigations** (you'll remember that this was formerly named **Opportunities**):



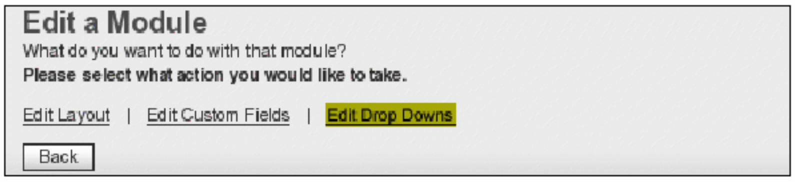
Welcome to Studio!

Please select a module from below.

Accounts | Bug Tracker | Campaigns | Contacts | Documents | Investigations | Invoices | Leads

Meetings | Notes | Preliminary Investigations | Project Tasks | Projects | Surveillance | Targets | Tasks

At this point you have to select **Edit Drop Downs** from the available options.



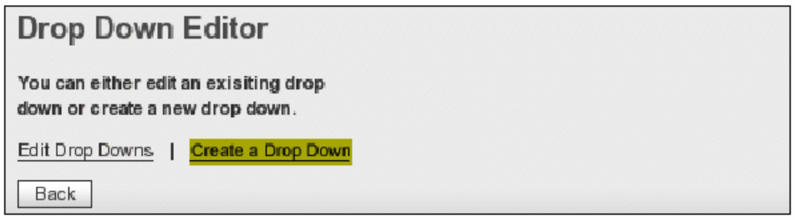
Edit a Module

What do you want to do with that module?

Please select what action you would like to take.

Edit Layout | Edit Custom Fields | Edit Drop Downs

Once you've done that you can edit existing dropdowns, but at the moment select **Create a Drop Down.**

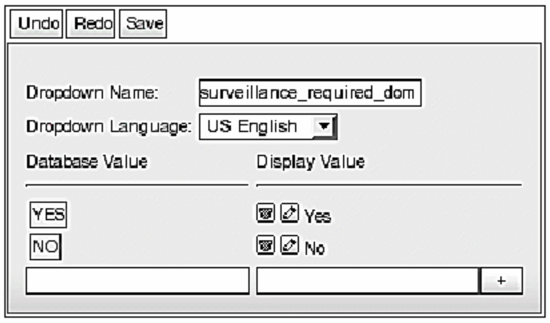


Drop Down Editor

You can either edit an exisiting drop down or create a new drop down.

Edit Drop Downs | Create a Drop Down

You can now create your new dropdown by giving it a suitable name, and assigning options to it. As you create each option you'll find that you need to enter two values— the value to be displayed on the screen and the value to stored in the database:

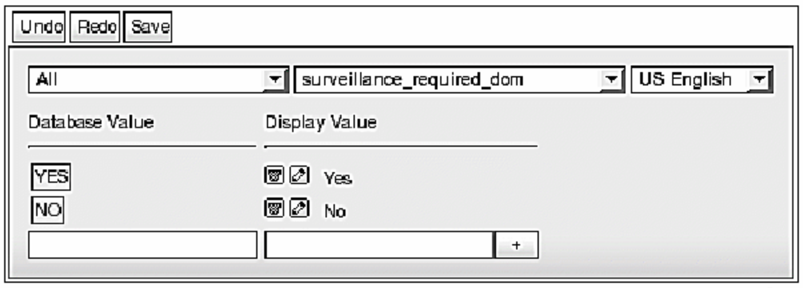


Dropdown Name: surveillance_required_dom

Dropdown Language: US English

Database Value	Display Value
YES	Yes
NO	No

Interestingly, once you've clicked **Save** you'll find that the dropdown is actually available to **All** modules, and not just **Preliminary Investigations**:



Now, all of that may seem a little long-winded—especially if you have a number of dropdowns to create. In fact, you will probably find that it is quicker and easier to create a dropdown manually, and so that's what we'll look at next.

Manually Adding a Drop-down Box

You will find that adding a dropdown manually is much simpler than adding one using the Studio—as long as you're happy to edit files. Well, only one file. You need to edit a file that you have already used in Chapters 1 and 2: `custom/include/language/en_us.lang.php`.

Now we're going to use `custom/include/language/en_us.lang.php` to create a new dropdown. All you have to do is to add the definition for the dropdown:

```
$app_list_strings['surveillance_required_dom'] = array (
    'YES' => 'Yes',
    'NO' => 'No',
  );
```

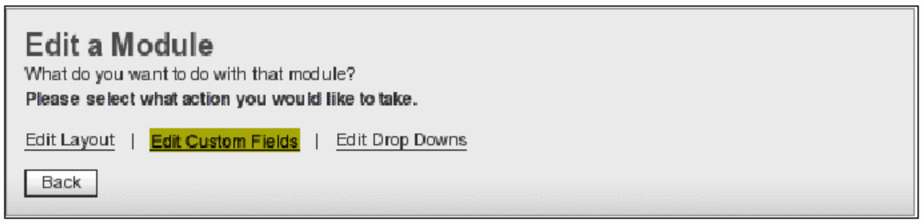
And that's all there is to creating a new dropdown. Next we need to look at the second stage of adding your custom field to a module tab—creating the custom field itself.

You have learned that we can create a dropdown either by using the Studio or by doing it manually. Custom field creation is exactly the same.

Using Studio to Create a Custom Field

Having defined a dropdown we now need to create a custom field. In actual fact this is a table column in the database. We will see how to do this manually, but first let's see how to use the Studio to do the job.

You'll remember that earlier we selected a module to edit (**Preliminary Investigations**) and then **Edit Drop Downs** – this time select **Edit Custom Fields**:

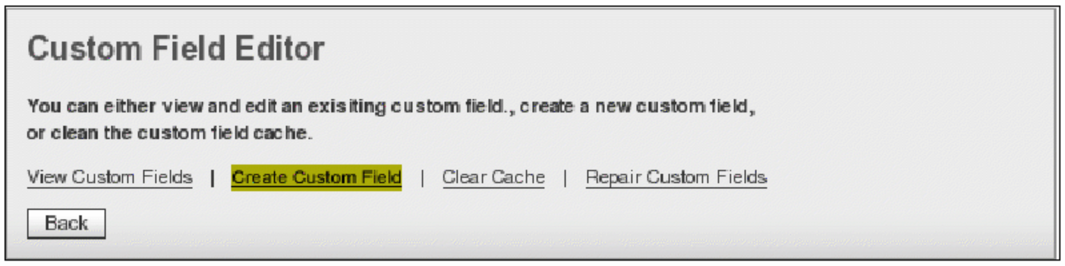


Edit a Module

What do you want to do with that module?

Please select what action you would like to take.

Next you'll need to click on **Create Custom Field**:

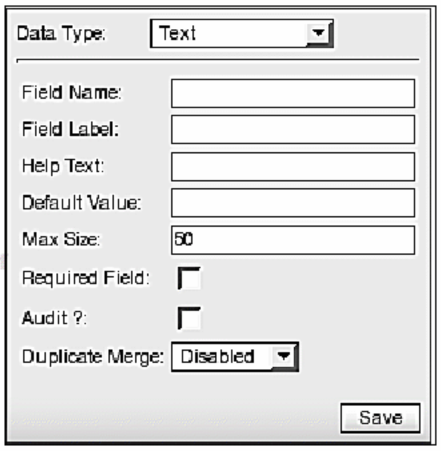


Custom Field Editor

You can either view and edit an exisiting custom field., create a new custom field, or clean the custom field cache.

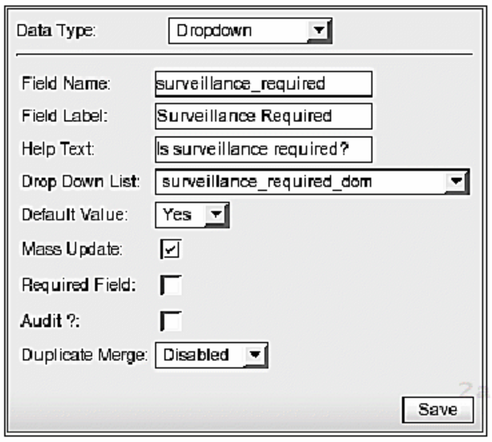
View Custom Fields | Create Custom Field | Clear Cache | Repair Custom Fields
Back

And then you'll be presented with the default field – the **Text** data type:



Data Type:	Text
Field Name:	
Field Label:	
Help Text:	
Default Value:	
Max Size:	50
Required Field:	☐
Audit ?:	☐
Duplicate Merge:	Disabled

Of course, we're not interested in the text field, we're interested in creating a dropdown. If you select **Dropdown** then you'll see that you'll be presented with a list of existing option lists. The one that we've already created should be available in this list:



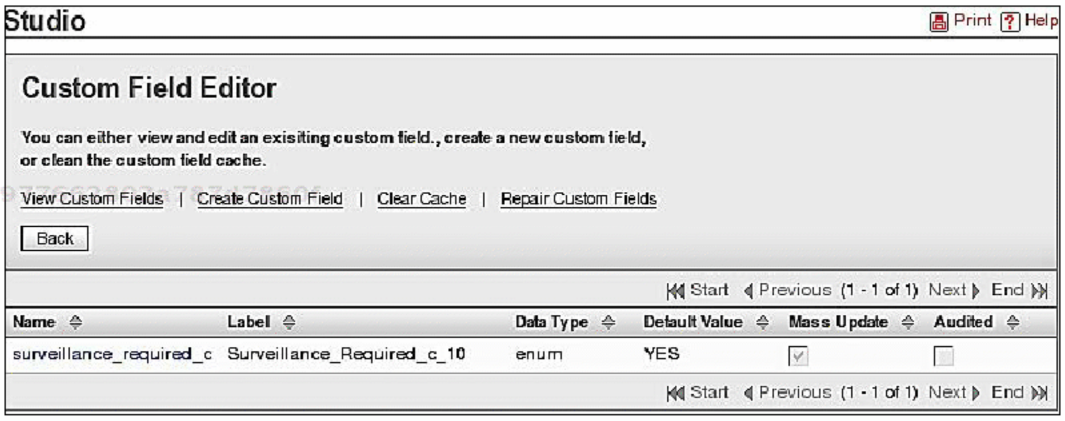
Data Type:	Dropdown
Field Name:	surveillance_required
Field Label:	Surveillance Required
Help Text:	Is surveillance required?
Drop Down List:	surveillance_required_dom
Default Value:	Yes
Mass Update:	✓
Required Field:	
Audit ?:	
Duplicate Merge:	Disabled

Save

Most of the fields are self explanatory, but it's worth just looking at a couple of them before we move on:

- **Mass Update** – If you enable this then you can include your field in the mass updates in the module tab screen.
- **Audit** – This tells SugarCRM to track any change that you make to your field.
- **Duplicate Merge** – This allows you to merge any duplicate records.

Once you've entered all of the details then press **Save**, and the Studio will update the SugarCRM database for you:



Studio

Custom Field Editor

You can either view and edit an exisiting custom field., create a new custom field, or clean the custom field cache.

View Custom Fields | Create Custom Field | Clear Cache | Repair Custom Fields

Back

Start Previous (1 - 1 of 1) Next End

Name	Label	Data Type	Default Value	Mass Update	Audited
surveillance_required_c	Surveillance_Required_c_10	enum	YES	✓	

Start Previous (1 - 1 of 1) Next End

You will notice that the details are not saved exactly as you entered them:

- **_c** is appended onto the end of your field name.
- **_c_10** is appended onto the end of your field label.

- Spaces in the field name and label are replaced by underscores.
- The data type is no longer shown as **Dropdown**, it is defined as **enum** – this is because there isn't a data type of 'dropdown'; the data is actually stored as an **enum** (i.e. a list).

Now you're ready to add the dropdown to your module tab. However, first you may want to consider how to create the field definitions manually.

Creating the Custom Field Manually

We've already established that a custom field is simply a reference stored in a database. All we have to do is to insert the appropriate information into the database ourselves:

```
insert into fields_meta_data
    (       id,
            name,
            label,
            help,
            custom_module,
            data_type,
            ext1,
            default_value,
            date_modified
    )
values
    (       'Opportunitiessurv1_req_c',
            'surv1_req_c',
            'surv1_req_c_10',
            'Surveillance?',
            'Opportunities',
            'enum',
            'surveillance_required_dom',
            'YES',
            now()
    )
;
```

There is, of course, a major advantage here, you can create a simple script to create all of the custom fields that you need – especially useful when you come to migrating from your development environment to your live environment (we'll discuss that further in Chapter 7).

Now, we're not quite finished with the database yet. If you've used Studio to add a custom field for the Opportunities module then you will find that you have a table named `opportunities_cstm` in the database (and don't forget—the renaming of the module is only for the browser—the SugarCRM structure remains the same). If not then you'll need to create it yourself. This table requires a new field for every custom field that you add (to `opportunities`, of course).

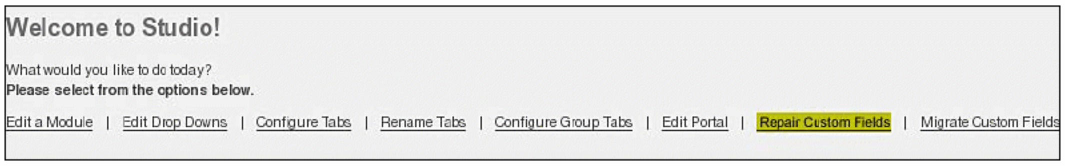
So, depending on whether `opportunities_cstm` exists or not, you'll need to do one of the following:

```
create table  opportunities_cstm (surv1_req_c varchar(150));
```

or:

```
alter table opportunities_cstm add surv1_req_c varchar(150);
```

On the other hand, once you've created your fields, then you may prefer to let the Studio set up `opportunities_cstm` for you. To do that just uses the Studio's **Repair Custom Fields** facility:

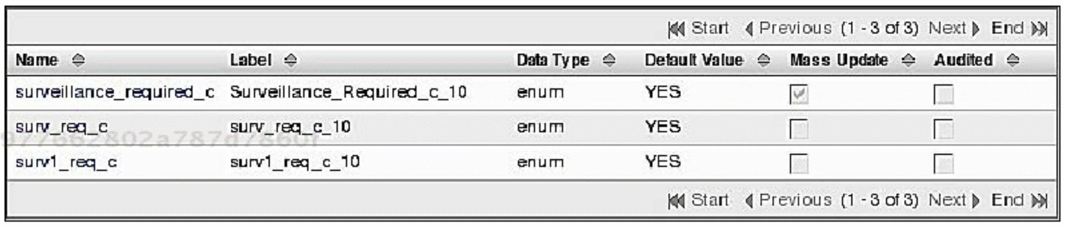


Welcome to Studio!

What would you like to do today?
Please select from the options below.

Edit a Module | Edit Drop Downs | Configure Tabs | Rename Tabs | Configure Group Tabs | Edit Portal | Repair Custom Fields | Migrate Custom Fields

Once you've added all of the fields that you want then you can view them via the Studio:



Start Previous (1 - 3 of 3) Next End

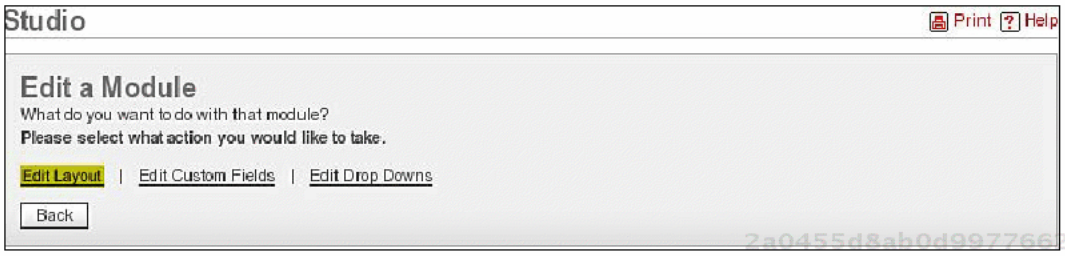
Name	Label	Data Type	Default Value	Mass Update	Audited
surveillance_required_c	Surveillance_Required_c_10	enum	YES	✓	☐
surv_req_c	surv_req_c_10	enum	YES	☐	☐
surv1_req_c	surv1_req_c_10	enum	YES	☐	☐

Start Previous (1 - 3 of 3) Next End

With your custom fields defined you can add them to your module tabs.

Adding the Dropdown to a Module Tab

You'll need to return to the module editor in the Studio in order to add your newly created custom field, but this time go to the **Edit Layout** link:

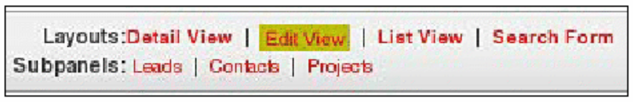


Studio

Edit a Module
What do you want to do with that module?
Please select what action you would like to take.

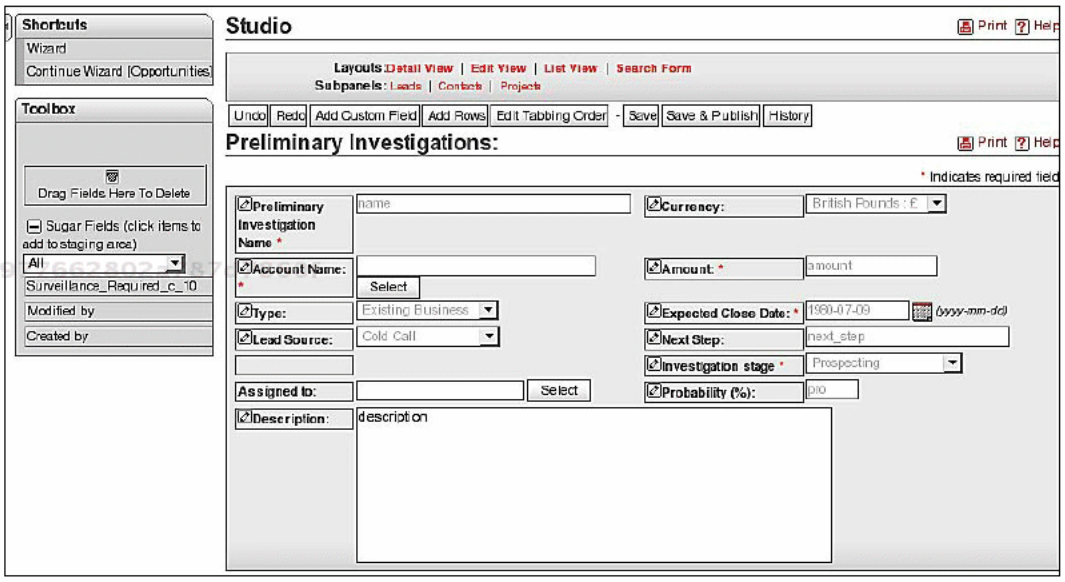
Edit Layout | Edit Custom Fields | Edit Drop Downs

At this point you'll be presented with the list the layouts that you're able to modify:



Layouts: Detail View | Edit View | List View | Search Form
Subpanels: Leads | Contacts | Projects

Now, I'm sure you'll agree that so far the development environment hasn't been exactly WYSIWYG. However, if you click on **Edit View** (for example), you'll find that you see the layout to be modified, and a toolbox:



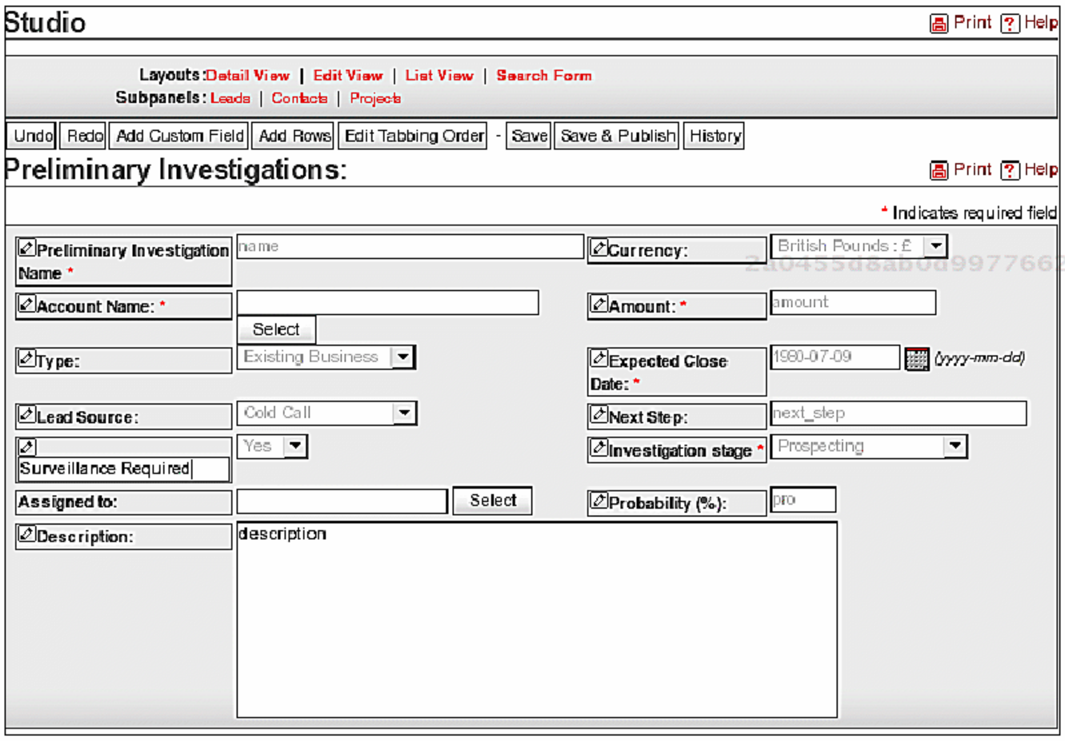
Shortcuts
Wizard
Continue Wizard [Opportunities]
Toolbox
Drag Fields Here To Delete
Sugar Fields (click items to add to staging area)
All
Surveillance_Required_c_10
Modified by
Created by
Studio
Print
Help
Layouts: Detail View | Edit View | List View | Search Form
Subpanels: Leads | Contacts | Projects
Undo
Redo
Add Custom Field
Add Rows
Edit Tabbing Order
Save
Save & Publish
History

Preliminary Investigations:

* Indicates required field

Preliminary Investigation Name *	name	Currency:	British Pounds : £
Account Name: *	Select	Amount: *	amount
Type:	Existing Business	Expected Close Date: *	1980-07-09 (yyyy-mm-dd)
Lead Source:	Cold Call	Next Step:	next_step
		Investigation stage *	Prospecting
Assigned to:	Select	Probability (%):	pro
Description:	description		

You'll see that the toolbox contains the field that we created, and you can drag and drop it into an appropriate location (you'll find a space between **Lead Source** and **Assigned to**):



Studio — Print Help

Layouts: Detail View | Edit View | List View | Search Form
Subpanels: Leads | Contacts | Projects

Undo | Redo | Add Custom Field | Add Rows | Edit Tabbing Order - Save | Save & Publish | History

Preliminary Investigations: — Print Help

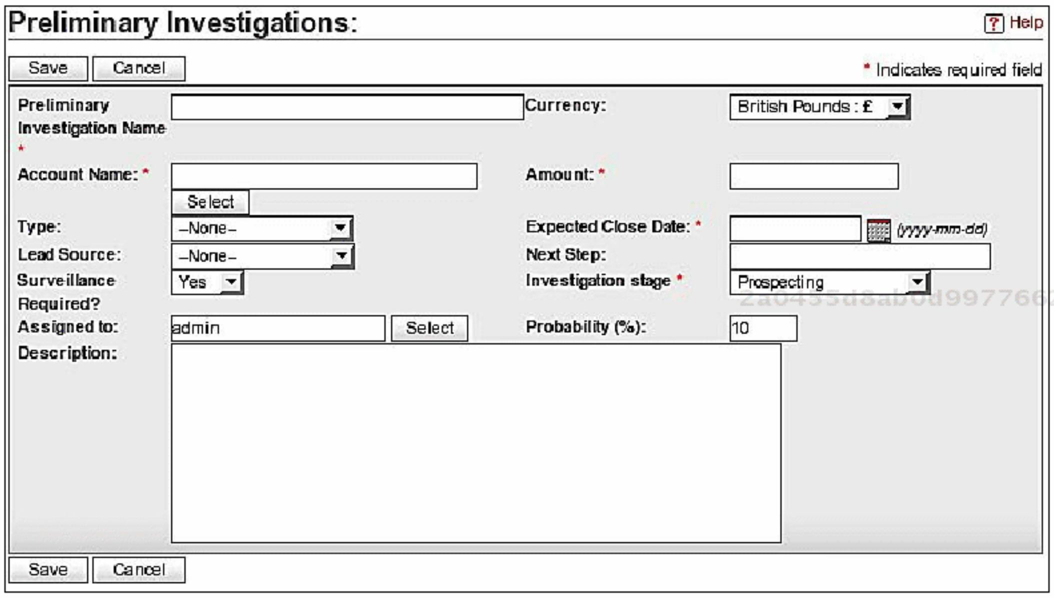
* Indicates required field

Preliminary Investigation Name *	name	Currency:	British Pounds : £
Account Name: *	Select	Amount: *	amount
Type:	Existing Business	Expected Close Date: *	1980-07-09 (yyyy-mm-dd)
Lead Source:	Cold Call	Next Step:	next_step
Surveillance Required	Yes	Investigation stage *	Prospecting
Assigned to:	Select	Probability (%):	pro
Description:	description		

All that is left to define is the text to be displayed next to your new drop-down box. You can either do this by using the Studio, or by adding a line to `custom/modules/Opportunities/ language/en_us.lang.php`:

```
$mod_strings['Surveillance_Required_c_10'] = "Surveillance Required?";
```

Once you've done all of that then you can click on **Save & Publish** to make the new layout available to all of your users:



Preliminary Investigations:

Help

Save Cancel

* Indicates required field

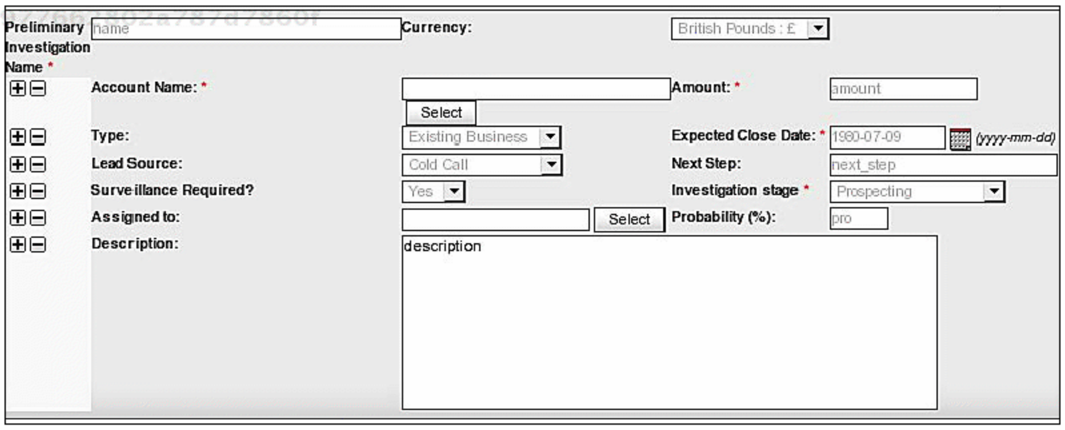
Preliminary Investigation Name *		Currency:	British Pounds : £
Account Name: *	Select	Amount: *	
Type:	–None–	Expected Close Date: *	(yyyy-mm-dd)
Lead Source:	–None–	Next Step:	
Surveillance Required?	Yes	Investigation stage *	Prospecting
Assigned to:	admin Select	Probability (%):	10
Description:			

Save Cancel

Of course that's fine for a single field (there's a gap for it), but you're going to have to make space for any additional fields.

Adding Rows

If you need to add more that one field then you'll need to add additional rows. You may have already noticed that one of the buttons on the layout screen is entitled **Add Rows** – click on this and you can add as many rows as you need:

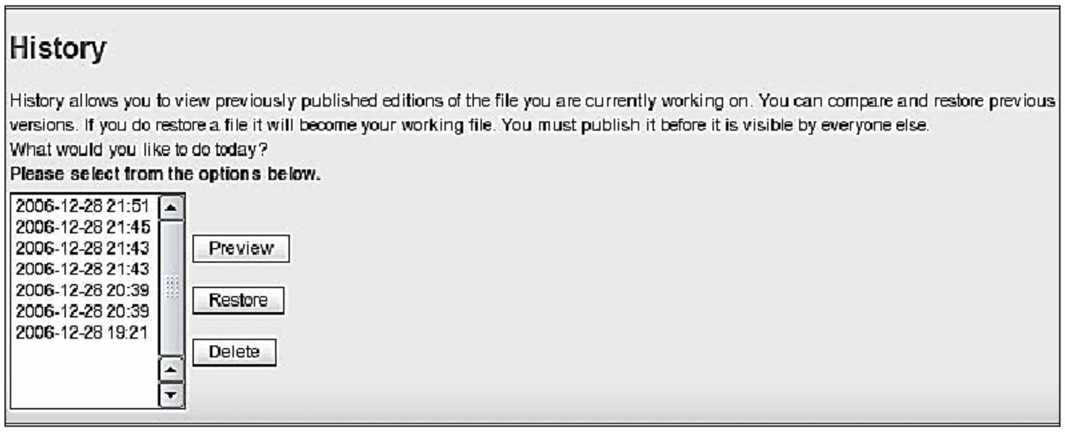


Preliminary Investigation Name *	name	Currency:	British Pounds : £
Account Name: *	Select	Amount: *	amount
Type:	Existing Business	Expected Close Date: *	1980-07-09 (yyyy-mm-dd)
Lead Source:	Cold Call	Next Step:	next_step
Surveillance Required?	Yes	Investigation stage *	Prospecting
Assigned to:	Select	Probability (%):	pro
Description:	description		

Adding rows is very easy (just press one of the **+** buttons). However, you need to be careful if you want to delete a row. If you accidentally delete a row containing important fields, you will find that there is no **Undo** button. You will also find that any fields that you delete by doing this do not appear in the toolbox when you return to the layout editing screen.

Recovering Previous Versions of a Layout

If, for any reason, you decide that you need to roll back to a previous layout then click on the **History** button. You can then view (and restore) the layout that you require:


History

History allows you to view previously published editions of the file you are currently working on. You can compare and restore previous versions. If you do restore a file it will become your working file. You must publish it before it is visible by everyone else.
What would you like to do today?
Please select from the options below.

2006-12-28 21:51
2006-12-28 21:45
2006-12-28 21:43
2006-12-28 21:43
2006-12-28 20:39
2006-12-28 20:39
2006-12-28 19:21

Manually Modifying Layouts

We've seen that we can use the Studio in order to modify the layout of the SugarCRM screens, but, as you'd expect, we can do this manually as well. The views that we can edit in the Studio are:

- Display
- Edit
- List
- Search

We've already established that the files for **Preliminary Investigations** are stored in the `modules/Opportunities` directory. All we have to do is to find the right files to edit. In the directory you'll find:

- `DisplayView.html`
- `EditView.html`

- `ListView.html`
- `SearchView.html`

If you have already modified the edit view, and now look at `EditView.html`, then you'll find that it contains something like:

```
<tr><!-- BEGIN: open_source -->
<td  class="dataLabel">
  <span sugar='slot11'>
    {MOD.Surveillance_Required_c_10}
  </span sugar='slot'>
</td>
<td class="dataField">
  <span sugar='slot11b'>
    <select title='{SURVEILLANCE_REQUIRED_C_HELP}'
    name="surveillance_required_c">{OPTIONS_SURVEILLANCE_REQUIRED_C}
    </select>
  </span sugar='slot'>
</td>
<td  class="dataLabel">
  <span sugar='slot12'>
    {MOD.LBL_SALES_STAGE}
    <span class="required">{APP.LBL_REQUIRED_SYMBOL}</span>
  </span sugar='slot'>
</td>
<td class="dataField">
  <span sugar='slot12b'>
    <select tabindex='2'
     name='sales_stage'>{SALES_STAGE_OPTIONS}
    </select>
  </span sugar='slot'>
</td>
<!-- END: open_source --></tr>
```

You'll realize that this is simple HTML code for adding lines to a table, but that it also makes use of some of our SugarCRM variables. Now, if we look back at the field that we manually created in this chapter then we'll see that its details are:

- name – surv1_req_c
- label – surv1_req_c_10

All we have to do is to add another line for our extra field at the end of the table of details in `EditView.html`:

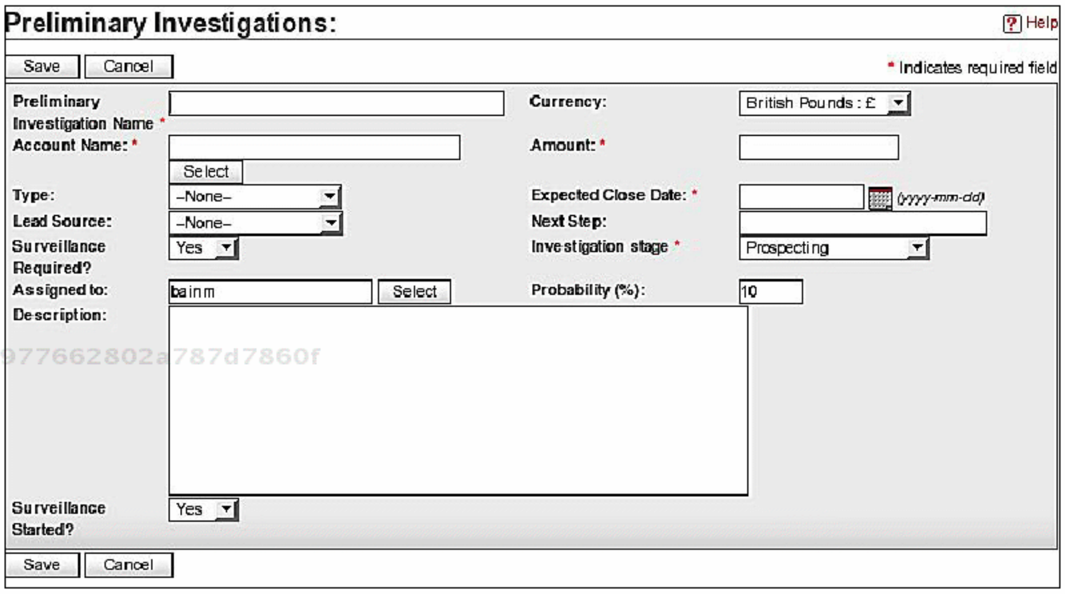
```
<tr><!-- BEGIN: open_source -->
<td  class="dataLabel">
```

```
      <span sugar='slot16'>
        {MOD.SURV1_REQ_C_10}
      </span sugar='slot'>
    </td>
    <td class="dataField">
      <span sugar='slot16b'>
        <select title='{SURV1_REQ_C_HELP}'
        name="surv1_req_c">{OPTIONS_SURV1_REQ_C}
        </select>
      </span sugar='slot'>
    </td>
    <!-- END: open_source --></tr>
```

Again, we're just using simple HTML to add a line to the table, but including references to the new field that we've created. And don't forget to modify `custom/modules/Opportunities/language/en_us.lang.php` to add a label for the dropdown:

```
    $mod_strings['surv1_req_c_10'] = "Surveillance Started?";
```

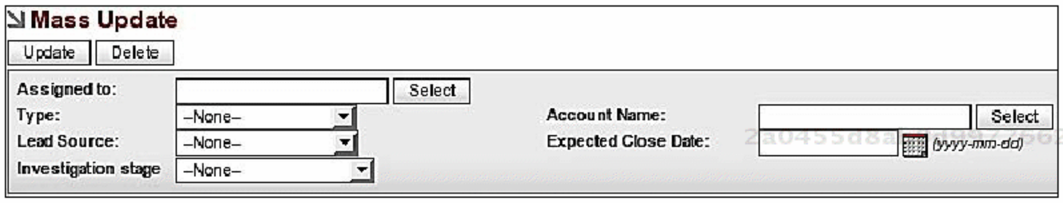
The end result is a screen containing two new dropdowns:



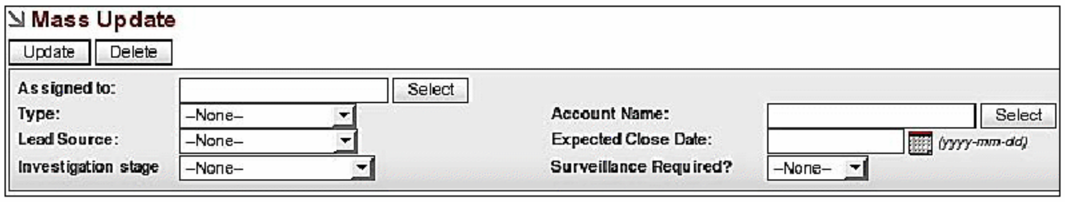
Preliminary Investigation Name *		Currency:	British Pounds : £
Account Name: *		Amount: *	
Type:	–None–	Expected Close Date: *	(yyyy-mm-dd)
Lead Source:	–None–	Next Step:	
Surveillance Required?	Yes	Investigation stage *	Prospecting
Assigned to:	bainm	Probability (%):	10
Description:			
Surveillance Started?	Yes		

Including Custom Fields in Mass Updates

I'm sure that you're already aware of the mass update function built into SugarCRM. This allows you to view a number of opportunities, cases, project tasks, etc., and then update key fields at the same time. So, for example, if you go to our **Primary Investigations** tab then you'll find that the default mass-update panel for the module looks like:



The mass-update function is very useful, and you will, of course, want to use your custom fields with it. In fact, if you've been following the examples in this chapter then you may find that one of the fields is already there:

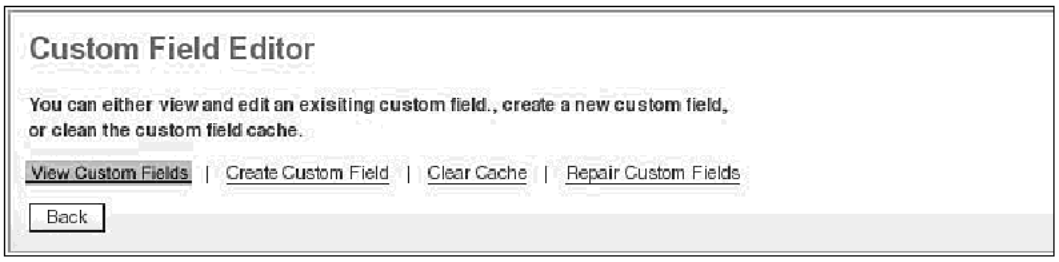


So, how do we add fields to the **Mass Update** sub-screen? Actually it is very easy. You may remember that earlier we saw how to create a new field using the Studio. On the Studio screen there's a box named **Mass Update** – tick this and your field will be automatically included in the sub-screen.

We have seen how to create the custom field manually using SQL, and, as you'd expect, it's just a matter of including a value for the appropriate field in the SQL statement:

```
insert into fields_meta_data
   (      id,
          name,
          label,
          help,
          custom_module,
          data_type,
          ext1,
          default_value,
            date_modified,
            mass_update
      )
  values
      (     'Opportunitiessurv2_req_c',
            'surv2_req_c',
            'surv2_req_c_10',
            'Surveillance?',
            'Opportunities',
            'enum',
            'surveillance_required_dom',
            'YES',
            now(),
            1
      )
  ;
```

But, what about fields that we've already created? Hopefully, you'll remember that back then we used the Studio to select the screen for creating new fields. This time select **View Custom Fields**:

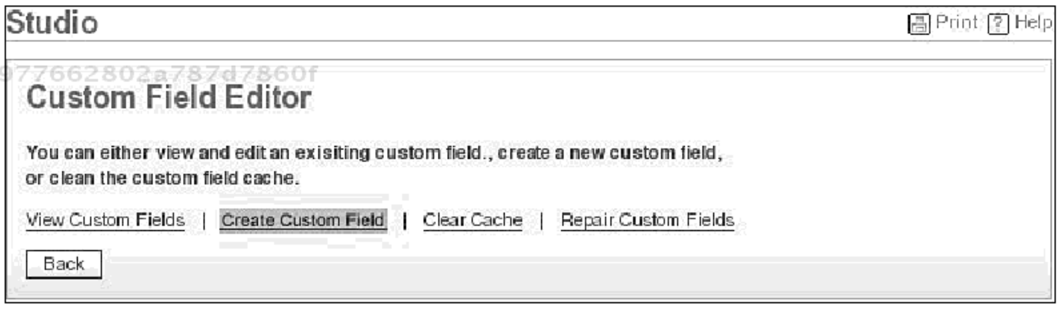


Custom Field Editor

You can either view and edit an exisiting custom field., create a new custom field, or clean the custom field cache.

View Custom Fields | Create Custom Field | Clear Cache | Repair Custom Fields

Back



Studio

Print Help

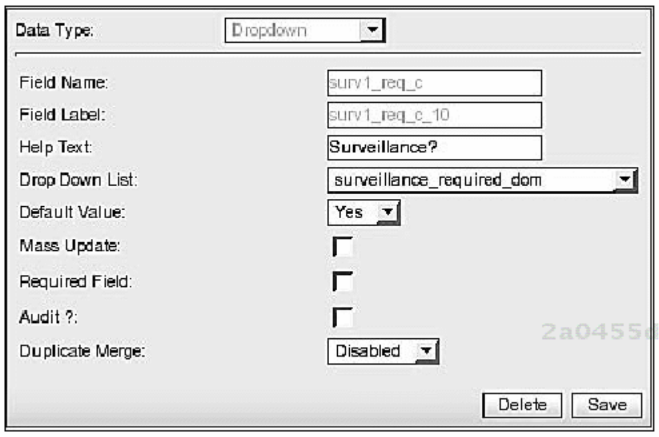
Custom Field Editor

You can either view and edit an exisiting custom field., create a new custom field, or clean the custom field cache.

View Custom Fields | Create Custom Field | Clear Cache | Repair Custom Fields

Back

Then you can choose the field that you're interested in and check the **Mass Update** box:



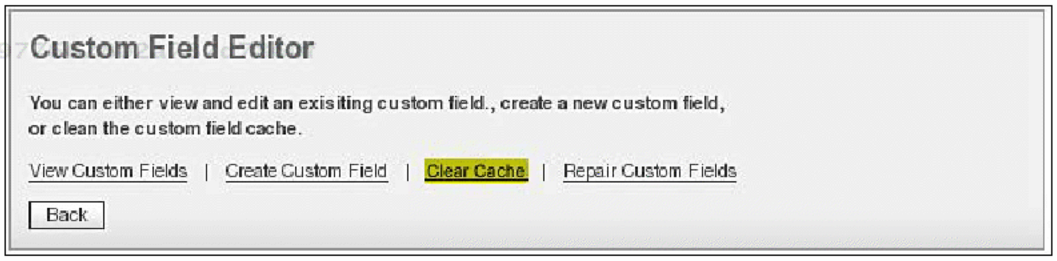
Data Type:	Dropdown
Field Name:	surv1_req_c
Field Label:	surv1_req_c_10
Help Text:	Surveillance?
Drop Down List:	surveillance_required_dom
Default Value:	Yes
Mass Update:	☐
Required Field:	☐
Audit ?:	☐
Duplicate Merge:	Disabled

Or, you can achieve the same by running SQL directly on the database:

```
update fields_meta_data
set mass_update=1
where id='Opportunitiessurv1_req_c';
```

Making Sure that Your Changes are Visible

Occasionally you'll make changes that aren't automatically passed through to all of your users. This is because SugarCRM uses a caching system for any custom fields (similar to, but not the same as, your web browser's caching). So, if you do change the *mass_update* field, make sure that you clear the cache via Studio's **Custom Field Editor**:



Custom Field Editor

You can either view and edit an exisiting custom field., create a new custom field, or clean the custom field cache.

View Custom Fields | Create Custom Field | Clear Cache | Repair Custom Fields
Back

Or you can do it manually by clearing the contents of the `cache/dynamic_fields` directory.

Limitations of the Mass Update

Now, before you run off and create loads of fields, it's worth noting that not all fields can be used for mass updating. The only field types that have these capabilities are:

- Dropdown (as we already know)
- Multiple Select
- Radio Buttons
- Date

That means that you *can't* use:

- Text
- Text Area
- Integer
- Decimal
- Checkbox
- Email
- Web Link
- HTML

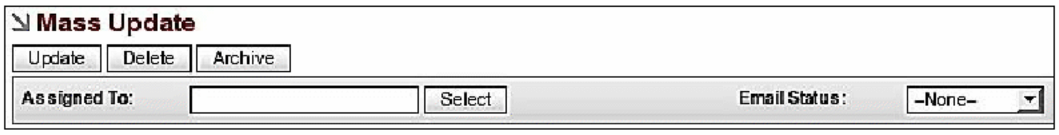
Adding Built-in SugarCRM Fields to the Mass Update

At this point you may be wondering if any other built-in fields can be added into the mass update. The answer is yes, but like custom fields not all types can be used. So, your next questions will be – which ones are they, and how do you do it?

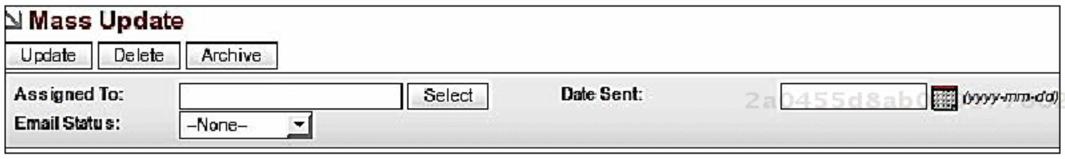
The built-in fields are handled differently to custom fields. In each module directory you'll find a `vardefs.php` file. Each `vardefs.php` file contains the details of fields to be used by the SugarCRM application. Any fields that can be used in the mass update have a *massupdate* property. Not all modules have fields that you can add to the mass update, but if you look in `modules/Emails/vardefs.php` (for example) then you'll find:

```
'date_start' => array (
                'name' => 'date_start',
                'vname' => 'LBL_DATE',
                'type' => 'date',
                'len' => '255',
                'rel_field' => 'time_start',
                'massupdate'=>false,
          ),
```

So, the standard mass update for emails looks like:



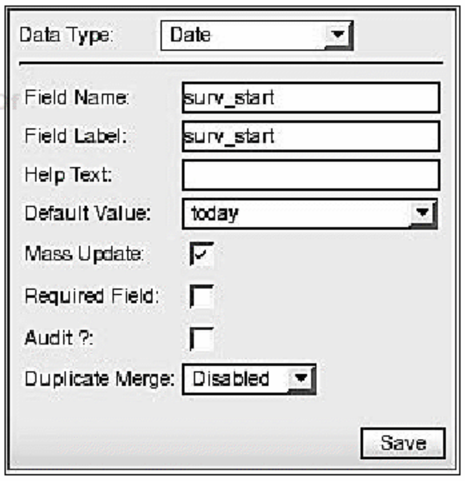
However, change `massupdate` to `true`, and you'll see:



As you can see the built-in fields have the same limitations as custom ones when it comes to mass updates—in this case `date_start` can only be included because it is a date field.

Creating other Field Types

We've seen how to create a drop-down field, both by using the Studio and manually, but you're probably wondering how to create other field types. Some (such as radio buttons) follow the same route as drop downs. Other types (such as dates) are simpler to create, since the process is the same except that you don't have to start by creating the drop-down box itself. So, if you want to create a date box (for instance) then go straight to the custom field editor:



Data Type:	Date
Field Name:	surv_start
Field Label:	surv_start
Help Text:	
Default Value:	today
Mass Update:	✓
Required Field:	
Audit ?:	
Duplicate Merge:	Disabled

From there on the process is exactly the same as creating the dropdown that we've already dealt with.

And if you're going to do this manually (using SQL) then you just need to know the data type—obviously in this case it would be *date*, so use an SQL insert query to do that:

```
insert into fields_meta_data
    (      id,
           name,
           label,
           help,
           custom_module,
           data_type,
           date_modified,
           mass_update
    )
values
    (      'Opportunitiessurv_start_c',
           'surv_start_c',
           'surv_start_c_10',
           'Surveillance Start',
           'Opportunities',
           'date',
           now(),
           1
    )
;
```

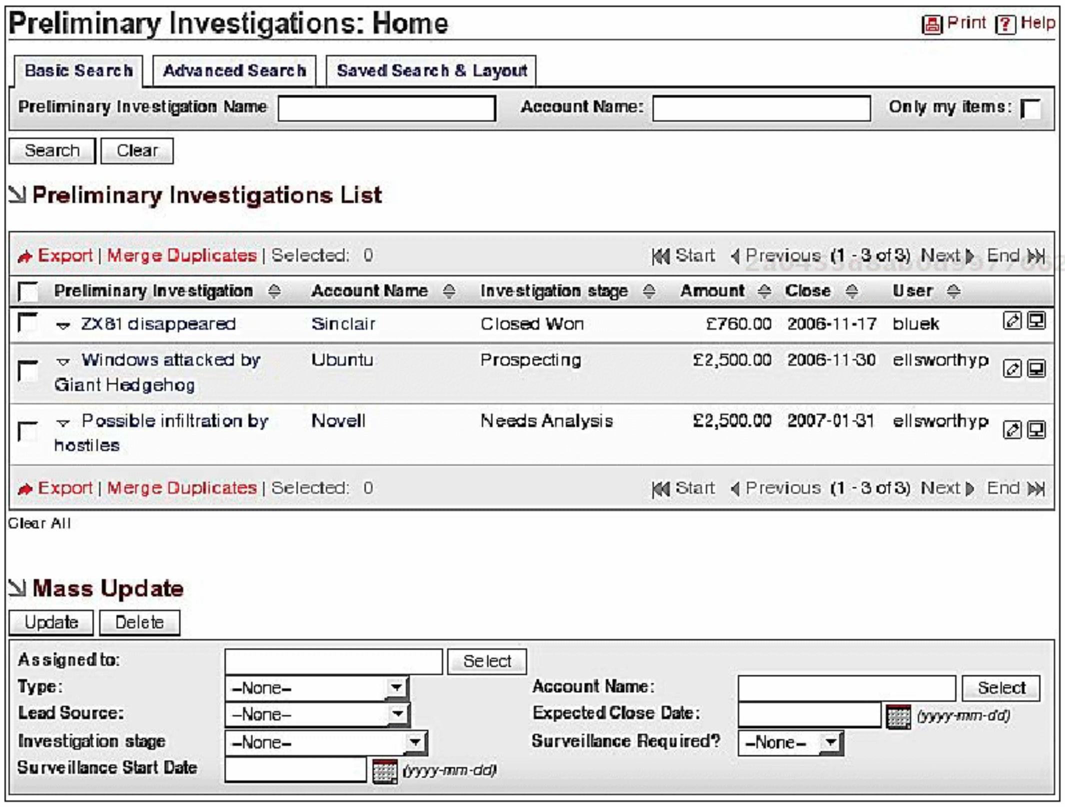
Having inserted the data, don't forget to add a new field to `opportunities_ctsm`, and a simple SQL alter statement will do that:

```
alter table opportunities_cstm add surv_start_c date;
```

Update `custom/modules/Opportunities/ language/en_us.lang.php`:

```
$mod_strings['surv_start_c_10'] = "Surveillance Start Date";
```

And once you've used the Studio or edited the `.html` files you'll have a date field on your screen as well as the dropdowns that we've already created – and Korora's life will suddenly become much easier:



Preliminary Investigations: Home

Basic Search | Advanced Search | Saved Search & Layout

Preliminary Investigation Name: | Account Name: | Only my items:

Preliminary Investigations List

Export | Merge Duplicates | Selected: 0 — Start Previous (1 - 3 of 3) Next End

Preliminary Investigation	Account Name	Investigation stage	Amount	Close	User
ZX81 disappeared	Sinclair	Closed Won	£760.00	2006-11-17	bluek
Windows attacked by Giant Hedgehog	Ubuntu	Prospecting	£2,500.00	2006-11-30	ellsworthyp
Possible infiltration by hostiles	Novell	Needs Analysis	£2,500.00	2007-01-31	ellsworthyp

Export | Merge Duplicates | Selected: 0 — Start Previous (1 - 3 of 3) Next End

Clear All

Mass Update

Assigned to:			
Type:	-None-	Account Name:	
Lead Source:	-None-	Expected Close Date:	(yyyy-mm-dd)
Investigation stage	-None-	Surveillance Required?	-None-
Surveillance Start Date	(yyyy-mm-dd)		

Field Data Types

So far we've looked at the dropdown (enum) and date field types. Now, if you are going to use the Studio to create your new fields then all you have to do is select the data type from the drop-down list. However, if you want to create the fields manually then you'll need to know what to enter in the `data_type` field in `fields_meta_data`:

- Text – varchar
- Text Area – text
- Integer – int
- Decimal – float
- Checkbox – bool
- Email – email

- Dropdown—enum
- Multiple Select—multienum
- Radio Buttons—radioenum
- Date—date
- Web Link—url
- HTML—html

With all this information at your fingertips you can now create whatever new fields your users require, and you can do it by using the methods that you feel most comfortable with—whether it be through the Studio, or by editing files and using SQL on the command line.

Summary

In this chapter we've seen how to create and make use of our own custom fields in SugarCRM modules. We've also seen how to include some of our fields (and some of the built-in SugarCRM fields) in the mass-update sub-panels.

We saw that the process for creating a custom field manually is the same as in the Studio; it's just that you'll be doing all of the things that the Studio would do for you.

In Chapter 4 we'll start to look at SugarCRM in more depth as we start to understand the structure of the application itself.