

1

Stamping Your Own Brand on SugarCRM

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

So, there you are, you've got SugarCRM up and running, you've played with it and maybe you've added an account, created a case, or a project, or maybe even sent an email or two. However, after a while you feel that SugarCRM is very *generic* (well, it has got to be hasn't it?) but generic is probably one thing that your business is not.

You will, of course, be wondering just what customizations can be carried out in order for SugarCRM to truly reflect the way in which your organization operates. To start with, we can group the possible customizations into:

- How SugarCRM looks
- How SugarCRM works

The first two chapters of this book are concerned with tailoring the looks of the tool, and if you're happy with SugarCRM's general look and feel, then just move to Chapter 3. You can always come back here once you've finished customizing the actual operation of SugarCRM.

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

However, if you do continue with this chapter (or if you have just come back to it) then by the end of Chapter 1 you'll be able to make SugarCRM look the way that you want it to — so that it reflects the image of your company, and uses the terminology of your users.

Before We Start...

I expect that one of the reasons that you've chosen SugarCRM is because it's open source, and any changes that you make are covered by the SugarCRM Public License (of course, only really need to worry, if you're planning to customize and then distribute SugarCRM). With that in mind, you may be wondering if there is anything

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

that you're *not allowed* to change. Only one. At the bottom of every screen you're expected to show the acknowledgement:



However, if you do want to read the full SugarCRM Public License you'll find it at <http://www.sugarcrm.com/crm/SPL>

Having identified what we're required to do under the SugarCRM Public License we can move on to choosing an example company. Throughout the book we'll be looking at Penguin P.I. – Private Investigators in the dark world between Windows and Linux.

When Penguin P.I.'s founder – Pygoscelis P. Ellsworthy – started the business it was a simple, two-man setup (or rather one man, one woman since he was working with the famous femme fatale, Korora Blue). However, after a few high profile cases he needed a bigger office, and it wasn't long before he had several offices spread across the world. It was at this point that he realized that a couple of spreadsheets weren't enough to manage all of his staff and his clients. Fortunately that's where SugarCRM comes in.

And that's our starting point. Now, you'll need a server complete with Apache, PHP, MySQL, and SugarCRM. How you do this is up to you, but you could:

- Follow the SugarCRM installation instructions – you'll find these at <http://www.sugarcrm.com/wiki/index.php?title=Installation>.
- Get hold of a decent book on the subject – such as *Implementing SugarCRM* by Michael J. R. Whitehead (ISBN 978-1-904811-68-8), published by Packt Publishing.
- Get someone else to do one of the previous steps for you – Pygoscelis, of course, he didn't actually do any of this *himself* – he got his new IT person, Robby Eudyptes, to do it for him.

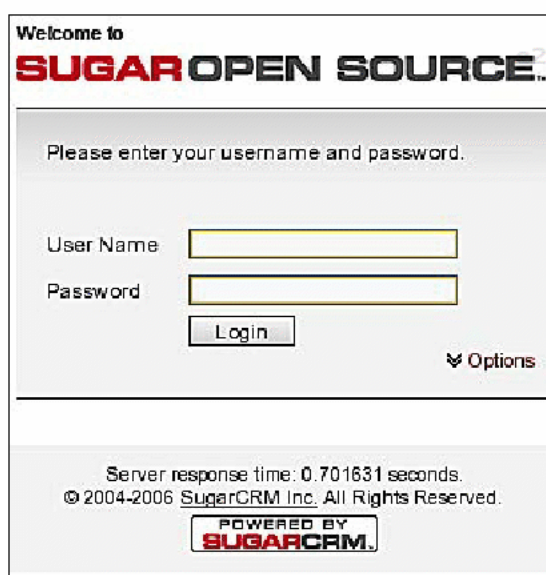
All of this assumes that you've given your servers good names – the most important job that any IT person can do. When Pygoscelis came to name all his servers he thought that he wouldn't follow the more common naming convention of choosing characters from J.R.R. Tolkien's *Lord of the Rings*. Instead he chose Homer's *The Iliad* (a fantastic book, and well worth reading if you get the chance – it shows you that human nature was just the same in 700BC as it is today). Therefore, he called his Linux servers Acamas, Aeneas, Cassandra, Hector, and Helenus. He also had one Windows server – Achilles (powerful, but with some major flaws).

With all of that done then you're ready to start customizing.

Customizing SugarCRM URL

At this stage we're not even going beyond SugarCRM's logon screen. If you've used an automatic installer (such as the very effective SpikeSource Windows Installer) then you'll be able open a browser, and type in the equivalent of `http://achilles/sugarcrm`. If you've manually installed SugarCRM on Linux then you'll probably need to type in something like: `http://hector/SugarOS-Full-4.5.0f`.

It doesn't matter if you're using Linux or Windows – in either case you'll be directed to the logon screen:



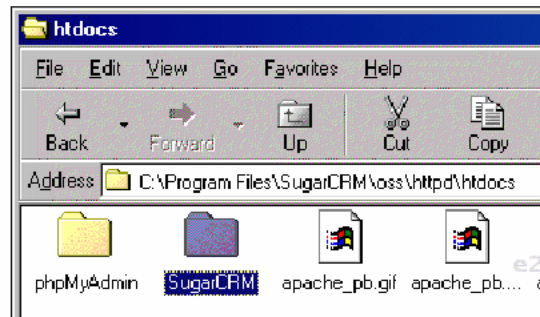
And you'll see that the URL reads: `http://achilles/sugarcrm/index.php?action=Login&module=Users` or `http://hector/SugarOS-Full-4.5.0f/index.php?action=Login&module=Users`. There's nothing actually wrong here, but the URLs don't really *tell* you much (apart from the fact that you're using SugarCRM), so it seems a good idea to change the URL to something more meaningful – something related to the actual project that you're working on. Fortunately this is a very easy change to make.

Changing the SugarCRM URL in Windows

If you're using Windows then the first thing that you'll have to do is to find the SugarCRM directory. Obviously if you've installed everything manually then you'll know where the directory is – however, if you've used the Windows installer then it may not be quite so obvious. You may (or may not) know that the SugarCRM

directory has to be in the document root for your web server, and if you've used the installer then this will probably be something like

C:\Program Files\SugarCRM\oss\httpd\htdocs:



e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

All you have to do is rename the directory (for instance our friend Pygoscelis might want it renamed `penguin_pi` for the Penguin P.I. organization), and then type the new URL into your browser (and for the Penguin P.I. organization this would be `http://achilles/penguin_pi`).

Changing the SugarCRM URL in Linux

In this instance Linux isn't too different from Windows – all you need to do is find the SugarCRM directory (again it will be in your web server's document root), and then rename it appropriately:

```
mv SugarOS-Full-4.5.0f penguin_pi
```

With that done the new URL will work:

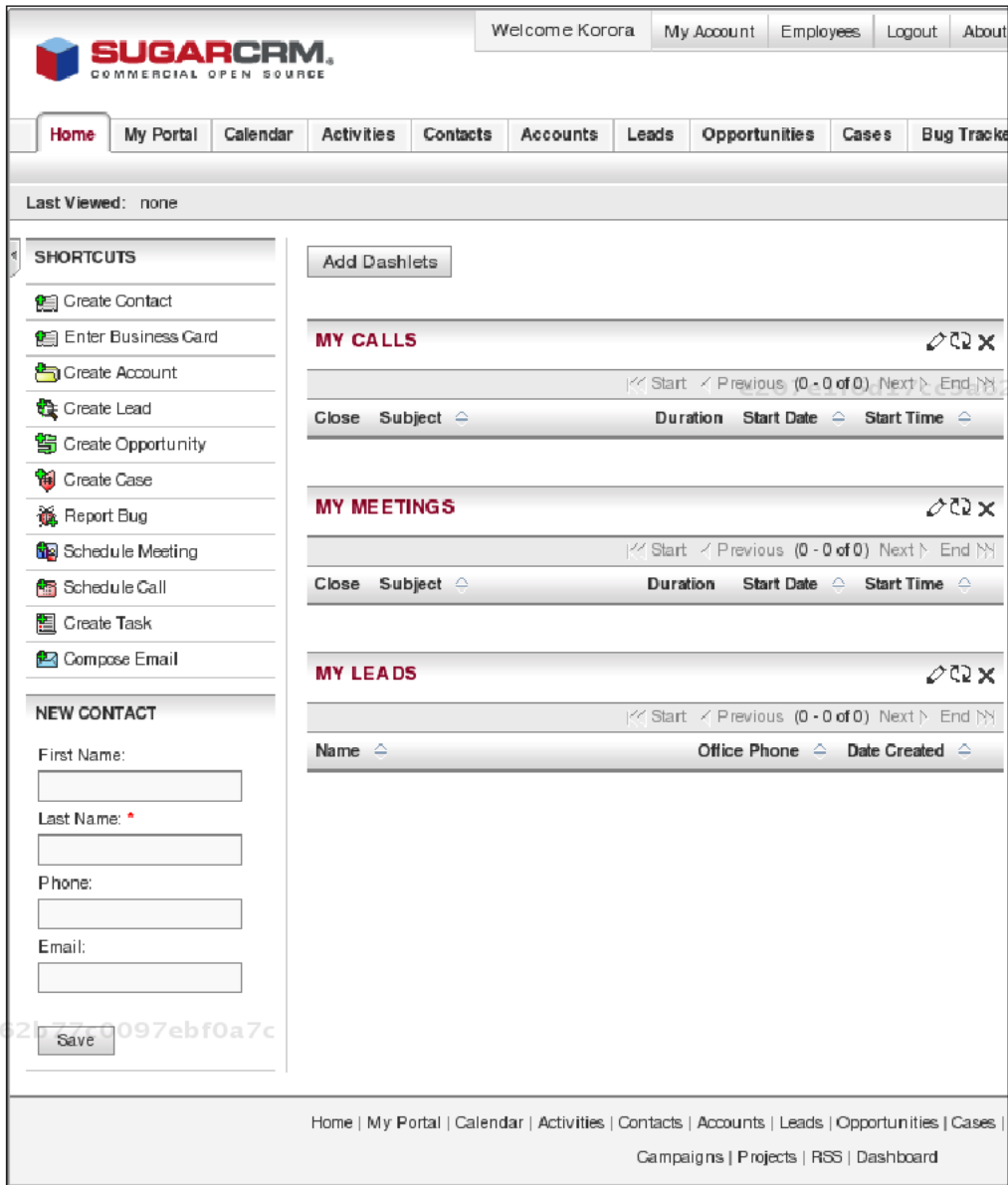
(e.g. `http://hector/penguin_pi/index.php?action=index&module=Home`)

Having sorted out SugarCRM's URL we can turn our attention to the rest of the screen.

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

Customizing SugarCRM Tabs

Your users are now able to access your SugarCRM implementation via a URL that means something to them. So, their first view of the system will be something like:



If we imagine Korora Blue looking at this for the first time, Korora would immediately realize that she can navigate around the system by using the SugarCRM Tabs. And she will recognize a lot of them (for example **Emails** and **Calendar**), but a lot will be new to her (for example **My Portal** and **Opportunities**).

So you've got two choices:

























- Re-train your staff so that they map their work to SugarCRM terminology.
- Re-name the SugarCRM tabs to the language of your organization.

Guess what we're going to start with.

Re-name the SugarCRM Tabs

To start with you need to log on to the administrator account (by default this will be by using the user name 'admin'), and then go to the **ADMINISTRATION: HOME** screen:

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

 ADMINISTRATION: HOME Print ? Help			
SUGAR NETWORK			
 Sugar Support Portal	Access your personalized portal for technical support and more	 Online Documentation	Get end-user and administrator documentation
 Sugar Updates	Check for latest updates.		
SYSTEM			
 System Settings	Configure system-wide settings	 Backups	Perform a backup
 Scheduler	Set up scheduled events	 Repair	Check and repair Sugar Suite
 Diagnostic Tool	Capture system configuration for diagnostics and analysis	 Currencies	Set up currencies and currency rates
 Upgrade Wizard	Upload and install Sugar Suite upgrades	 Module Loader	Add or remove Sugar modules, themes, and language packs
 Locale Settings	Set default localization settings for your system.		
USERS			
 User Management	Manage user accounts and passwords	 Role Management	Manage role membership and properties
EMAIL			
 Email Settings	Configure email settings	 Manage Email Queue	Manage the outbound email queue
 Inbound Email	Set up mailboxes to be monitored for inbound email		
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		
BUG TRACKER			
 Releases	Manage releases and versions		

Here you can either click **Rename Tabs**, or you can go to the **Studio** screen first:

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](#)

Now you can rename any of the SugarCRM tabs to something more appropriate to your organization:

UndoRedoSave

AllmoduleListUS English

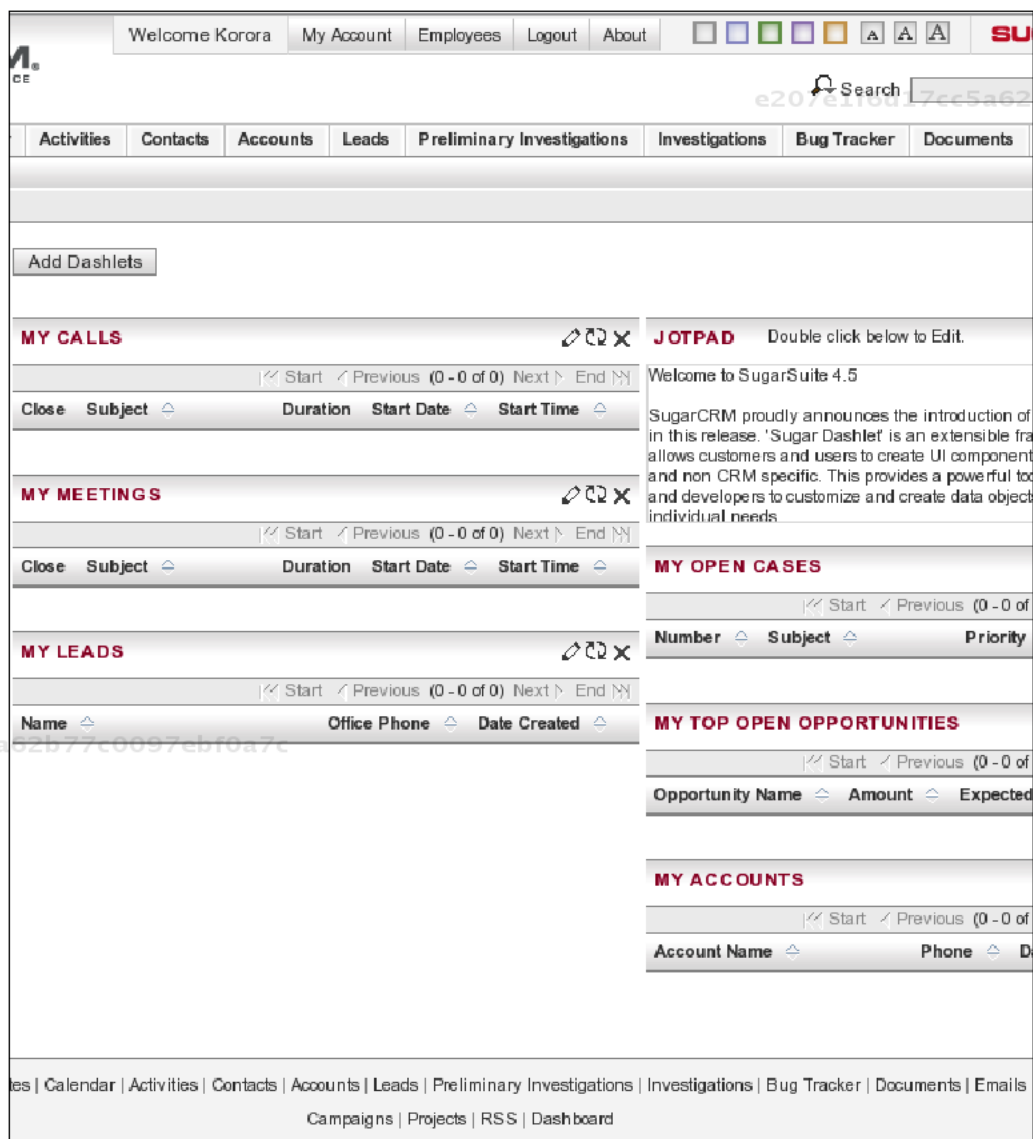
Database Value	Display Value
Home	Home
Dashboard	Dashboard
Contacts	Contacts
Accounts	Accounts
Opportunities	Opportunities
Cases	Investigations
Notes	Notes
Calls	Calls
Emails	Emails
Meetings	Meetings
Tasks	Tasks
Calendar	Calendar
Leads	Leads
Activities	Activities
Bugs	Bug Tracker
Feeds	RSS
IFrames	My Portal
TimePeriods	Time Periods
Project	Projects
ProjectTask	Project Tasks
Campaigns	Campaigns
Documents	Documents
Sync	Sync
Users	Users
Releases	Releases
Prospects	Targets
Queues	Queues
EmailMarketing	Email Marketing
EmailTemplates	Email Templates
ProspectLists	Target Lists
SavedSearch	Saved Searches
Invoice	Invoices

Stamping Your Own Brand on SugarCRM

Once you've saved your changes your users will see something that relates to them. For example, here we've renamed:

- My Portal => **Web Sites**
- Opportunities => **Preliminary Investigations**
- Cases => **Investigations**

And now Korora would have tabs that she'd immediately understand:



However, the screen still contains a large amount of the default text (which has nothing to do with Korora's job), and this would be even more apparent if Korora were to click on one of the newly named tabs such as **Preliminary Investigations**:

PRELIMINARY INVESTIGATIONS: HOME

Basic Search | Advanced Search | Saved Search & Layout

Preliminary Investigation Name: Account Name:

Search Clear

PRELIMINARY INVESTIGATIONS LIST

Export Merge Duplicates | Selected: 0 | Start Previous (1 - 3)

<input type="checkbox"/> Preliminary Investigation	Account Name	Investigation stage	Amount	Close
<input type="checkbox"/> ZX81 disappeared	Sinclair	Closed Won	£760.00	2006-11-17
<input type="checkbox"/> Windows attacked by Giant Hedgehog	Ubuntu	Prospecting	£2,500.00	2006-11-30
<input type="checkbox"/> Possible infiltration by hostiles	Novell	Needs Analysis	£2,500.00	2007-01-31

Export Merge Duplicates | Selected: 0 | Start Previous (1 - 3)

Clear All

MASS UPDATE

Update Delete

Assigned to: Select

Type:

Lead Source:

Investigation stage:

Account Name:

Expected Close Date:

Still not exactly 'custom' is it? And I'm sure that you've had a look around the administration screens to change other details on the screen, but not had any success. That's because any further changes need to be done outside the SugarCRM application, in the SugarCRM custom directory.

The SugarCRM Custom Directory

We'll do most of our work in the SugarCRM `custom` directory – in fact we've been using it already. You'll find it in your SugarCRM folder on your web server, and if you take a look there, you will see that there's already a file in the folder – when we saved the changes to the tab names SugarCRM created `custom/include/language/en_us.lang.php`. Examination of the file will reveal the changes made:

```
<?php
$app_list_strings['moduleList'] = array (
    'Home' => 'Home',
    'Dashboard' => 'Dashboard',
    'Contacts' => 'Contacts',
    'Accounts' => 'Accounts',
    'Opportunities' => 'Preliminary Investigations',
    'Cases' => 'Investigations',
    'Notes' => 'Notes',
    'Calls' => 'Surveillance',
    'Emails' => 'Emails',
    'Meetings' => 'Meetings',
    'Tasks' => 'Tasks',
    'Calendar' => 'Calendar',
    'Leads' => 'Leads',
    'Activities' => 'Activities',
    'Bugs' => 'Bug Tracker',
    'Feeds' => 'RSS',
    'iFrames' => 'Web Sites',
    'TimePeriods' => 'Time Periods',
    'Project' => 'Projects',
    'ProjectTask' => 'Project Tasks',
    'Campaigns' => 'Campaigns',
    'Documents' => 'Documents',
    'Sync' => 'Sync',
    'Users' => 'Users',
    'Releases' => 'Releases',
    'Prospects' => 'Targets',
    'Queues' => 'Queues',
    'EmailMarketing' => 'Email Marketing',
    'EmailTemplates' => 'Email Templates',
    'ProspectLists' => 'Target Lists',
    'SavedSearch' => 'Saved Searches',
    'Invoice' => 'Invoices',
);
?>
```

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruarye207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

And that should give you a clue as to how language customization works within SugarCRM.

Customizing the Text within SugarCRM Tab Screens

We're not going to look at *every* tab screen – once you know how to modify one then you can make changes as required. Since we've already been looking at the **Opportunities** screen let's carry on with that – but don't forget that the techniques we use will apply on any other screen that you want to customize.

We've seen that we use a file called `en_us.lang.php` in order to make our own changes to the text displayed on the SugarCRM screen, and you may well reason that we'll use this file to make changes to *any* text on the screen. Well, you're nearly right. We *will* use an `en_us.lang.php` file – *but not the same one*. This time we're going to use a file called `custom/modules/Opportunities/language/en_us.lang.php` (remember that `custom` is in your SugarCRM directory – and you'll need to create the sub-directories and the file itself).

The way in which this works is quite simple – SugarCRM will look at its default language files for the text to be displayed. However, if you've got your own definitions in a custom file then it will use those instead. Of course, next you'll need the default definitions so that you know what to put into your custom language file.

You may have already worked this out (but in case you haven't) the default language file for 'Opportunities' is `modules/Opportunities/language/en_us.lang.php`. If you examine the file then you'll find that it contains similar data to the `en_us.lang.php` that we've already seen:

```
$mod_strings = array (
    'LBL_MODULE_NAME' => 'Opportunities',
    'LBL_MODULE_TITLE' => 'Opportunities: Home',
    'LBL_SEARCH_FORM_TITLE' => 'Opportunity Search',
    'LBL_VIEW_FORM_TITLE' => 'Opportunity View',
    'LBL_LIST_FORM_TITLE' => 'Opportunity List',
    'LBL OPPORTUNITY_NAME' => 'Opportunity Name:',
```

As you can see, it contains the text to be displayed in the tab screen labels, and you may be wondering why we don't just edit this file instead. There are a few reasons:

- This gives you a nice fall-back position—if everything goes wrong then you just need to delete the custom files to return to the defaults.
- There are some lines in the default file that *must not* be changed—the application may get *very* upset with you if you do change them.
- You only need to worry about maintaining your own changes and not the whole file.
- If you use the custom files then your changes remain isolated from the core functionality while allowing you to make the customizations that you need.

So, the next stage is to edit the `custom/modules/Opportunities/language/en_us.lang.php` file so that it contains the text that you actually want to be displayed on the **Opportunities** screen:

```
<?php
$sopp_single = 'Preliminary Investigation';
$sopp_title = $sopp_single . 's';
$mod_strings['LBL_MODULE_NAME'] = $sopp_title;
$mod_strings['LBL_MODULE_TITLE'] = $sopp_title . ': Home';
$mod_strings['LBL_LIST_FORM_TITLE'] = $sopp_title . ' List';
$mod_strings['LBL OPPORTUNITY_NAME'] = $sopp_single . ' Name';
$mod_strings['LBL_NEW_FORM_TITLE'] = 'Create ' . $sopp_title;
$mod_strings['LNK_NEW OPPORTUNITY'] = 'Create ' . $sopp_title;
$mod_strings['LNK OPPORTUNITY LIST'] = $sopp_title;
$mod_strings['LBL_TOP OPPORTUNITIES'] = 'My Top Open ' . $sopp_title;
$mod_strings['LBL_DEFAULT_SUBPANEL_TITLE'] = $sopp_title;
?>
```

Notice that we haven't just hard coded all of the label details—this time we're using some variables (`$sopp_single` and `$sopp_title`). This means, of course, that if you decide to change the name of the module then you only need to change one line of code to update all the labels.

Once you've saved the file you can see the effects immediately by going to the **Opportunities** tab:

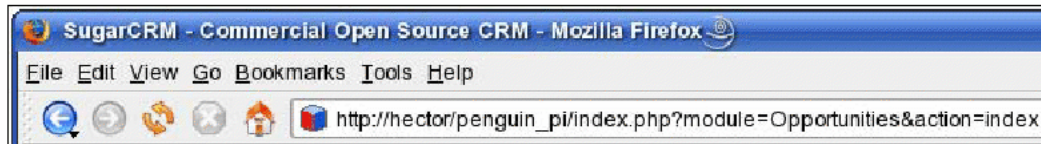
The screenshot displays the SugarCRM 'Opportunities' tab. At the top, there's a navigation bar with tabs: 'Activities', 'Contacts', 'Accounts', 'Leads', 'Preliminary Investigations' (selected), 'Investigations', 'Bug Tracker', and 'Documents'. Below this is a search bar and a 'Basic Search' section with fields for 'Opportunity Name' and 'Account Name'. The main section is titled 'OPPORTUNITY LIST' and contains a table with columns: 'Opportunity', 'Account Name', 'Sales Stage', 'Amount', and 'Close'. The table lists three opportunities: 'ZX81 disappeared' (Closed Won, £760.00, 2006-11-17), 'Windows attacked by Giant Hedgehog' (Prospecting, £2,500.00, 2006-11-30), and 'Possible infiltration by hostiles' (Needs Analysis, £2,500.00, 2007-01-31). Below the table is a 'MASS UPDATE' section with fields for 'Assigned to', 'Type', 'Lead Source', 'Sales Stage', 'Account Name', and 'Expected Close Date'.

Opportunity	Account Name	Sales Stage	Amount	Close
ZX81 disappeared	Sindair	Closed Won	£760.00	2006-11-17
Windows attacked by Giant Hedgehog	Ubuntu	Prospecting	£2,500.00	2006-11-30
Possible infiltration by hostiles	Novell	Needs Analysis	£2,500.00	2007-01-31

Obviously you need to repeat this process for each of the tab screens, but before long you won't have a generic CRM system – you will have a CRM system that uses the same terminology as your organization.

Changing the Browser Title

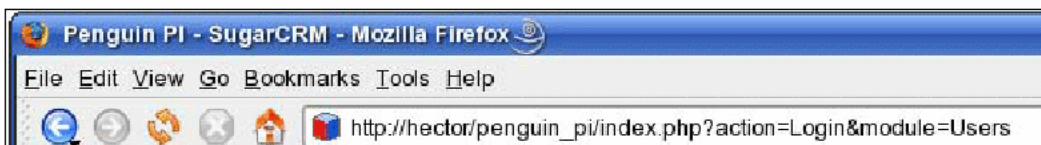
We've now modified the SugarCRM screens so that any users recognize the language, but we can still do more to reflect your company's brand. The first thing to look at is the browser title. At the moment it will look like:



While we're not trying to cover up the fact that we're using SugarCRM, we are trying to stamp our own brand onto the application. To do this we'll need to edit the `custom/include/language/en_us.lang.php` file again, and add a line:

```
$app_strings['LBL_BROWSER_TITLE'] = 'Penguin PI - SugarCRM';
```

Now, you need to refresh the browser:



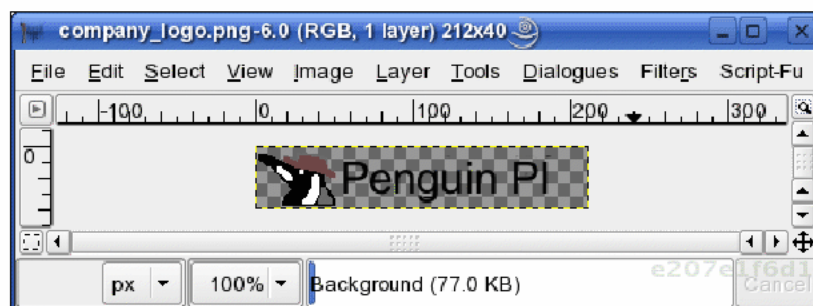
With the title set correctly, we can think about one of the most important parts of a brand – the company logo.

Adding a Company Logo

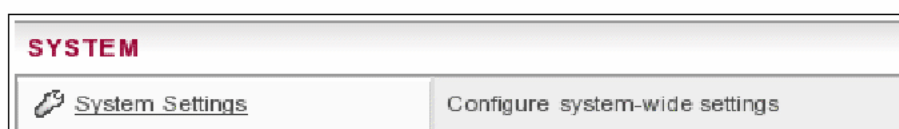
You may already have a logo that you're wanting to use with your SugarCRM installation. However, whether you're going to use an existing one, or you're creating a completely new one, there are a couple of things that you need to keep in mind:

- To be consistent with the SugarCRM layout your company logo must be 220x40 pixels.
- Your company logo should have a transparent background – so that it can work with different SugarCRM themes.

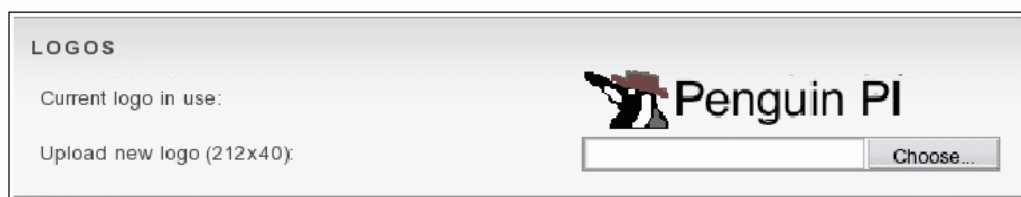
Then it's just a matter of creating the image with an appropriate piece of software; for example, you could use Gimp (Gimp is the GNU Image Manipulation Program, and if it's not already on your computer then you can get it from <http://www.gimp.org>).



Once you have saved your image you'll want to apply it to your SugarCRM implementation – so, it's back to the main administration screen, where you need to click on **System Settings**:



And then you can upload the logo:

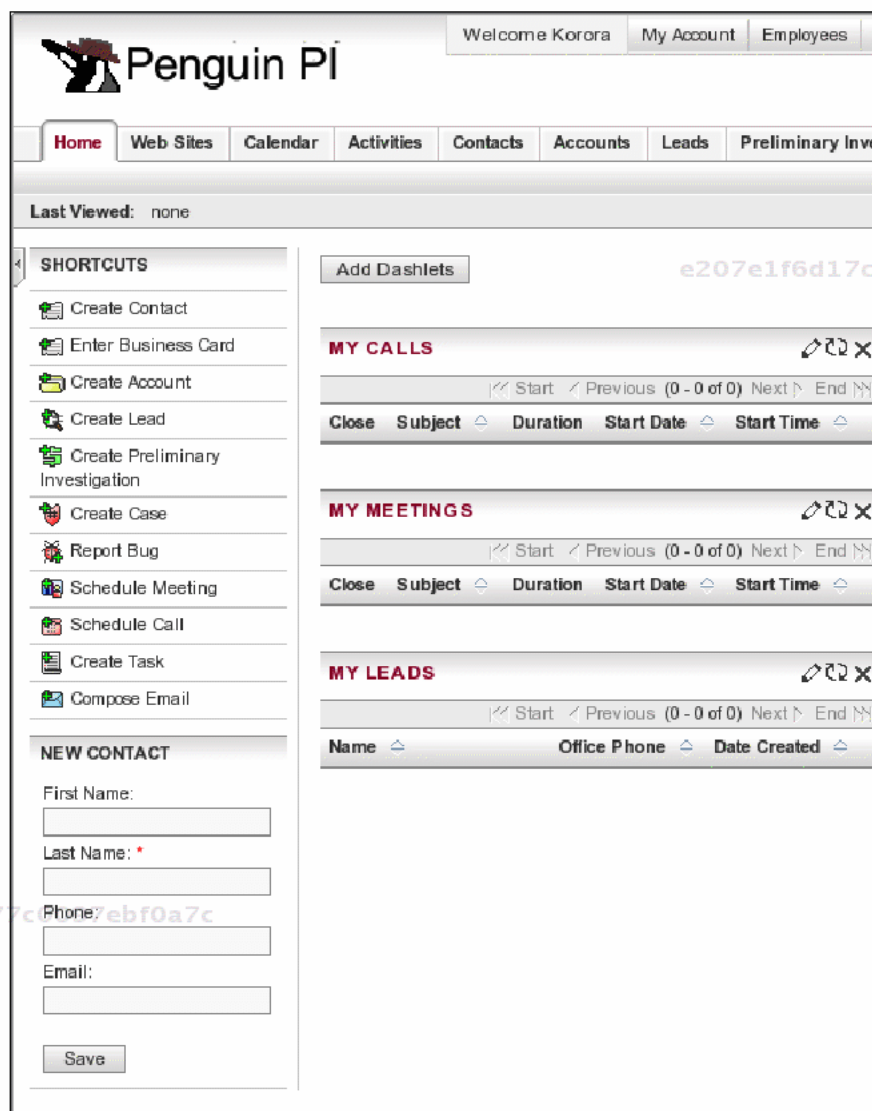


With that done you've got a SugarCRM implementation with:

- Tabs with titles that mean something to people in your organization
- Screens that use the correct terminology for your users
- A logo that illustrates the implementation belongs to your company

Stamping Your Own Brand on SugarCRM

Now your users can log on and see your company's SugarCRM application, and they'll know that it's something designed for them, and (hopefully) they'll be happy to use it:



At this stage you may decide that you don't need to do any further customizations to the look and feel of SugarCRM, and if that's the case then it's time to move on to Chapter 2 where we start looking at customizing the application contents. However, before you do that it's worth spending a little time considering another aspect of the SugarCRM front-end user experience – Themes.

Customizing SugarCRM Themes

We've spent time getting the general look and feel right, but (as you probably already know) your users can already customize SugarCRM by using themes – sometimes with quite startling results:



2b77c0097ebf0a7c
ebruary

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

If you're striving to create an application with your own brand then some of the themes may not work with your view of how things should look. If that's the case then you can:

- Create your own theme(s)
- Limit the themes that can be accessed by your users

So, that's what we'll look at next. However, before we start have a look at this:



You should see two lines, each saying **Hello Korora**. However, if any of your users are color blind then they may not be able to read the information (personally, I've got a red-green deficiency that means that I can read both lines but they give me a blinding headache).



The point is, of course, if you are going to be creating your own color scheme then be aware of the effect that it will have on your users. The same goes for the size of fonts that you want to use.

Creating a New Theme

You'll find that the easiest way to create a new theme is simply to copy an existing one and then modify it. Start by looking for the themes directory – a sub-directory of your main SugarCRM folder. On Linux you can achieve this by typing something like:

```
cd /var/www/htdocs/penguin_pi/themes
cp -R SugarClassic PenguinPI
```

or on Windows:

```
cd "C:\Program Files\SugarCRM\oss\httpd\htdocs\SugarCRM\themes"
copy sugarclassic penguinpi
```

You now need to move to your new directory, edit the config.php file, and add the theme name and description:

```
<?php
if(!defined('sugarEntry') || !sugarEntry) die('Not A Valid Entry
Point');
$theme_name = "Penguin PI";
$theme_description = "Penguin PI theme";
```

```
$version_compatibility = 3.0;  
$max_tabs = 12;  
$png_support = false;  
?>
```

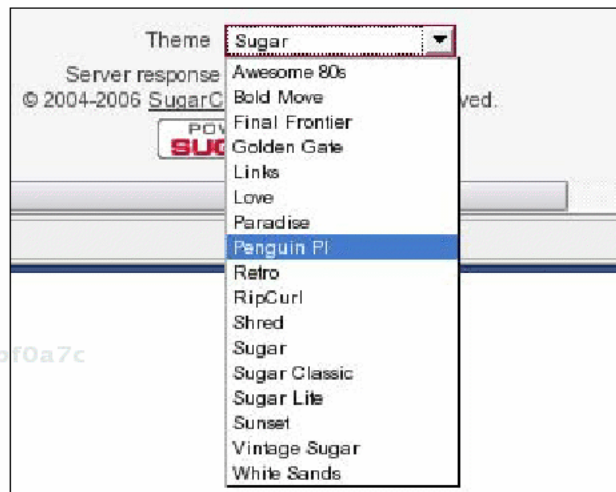
Also in the directory you'll find all of the files needed for formatting the theme itself. For now we'll just look at a very simple modification – we'll change the theme so that any link is highlighted in yellow when the mouse pointer is placed over it.

All you need to do is edit the `style.css` file in your new theme folder, and make the following change:

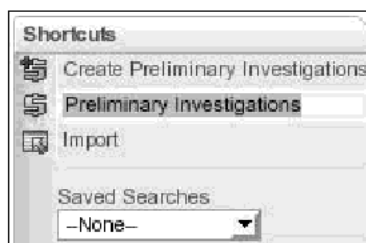
```
a:hover {  
    color: #666666;  
    text-decoration: underline;  
    background-color:yellow; //highlight the link  
}
```

e207e1f6d17cc5a62b77c0097ebf0a7c
ebrary

You'll probably need to restart your browser, but once you have done so, then you'll be able to select your new theme from the drop-down list at the bottom of the SugarCRM screen:

e207e1f6d17cc5a62b77c0097ebf0a7c
ebrary

This new theme will appear exactly the same as the **Sugar** theme, except that links will be highlighted in yellow when you place the mouse pointer over them:



Of course, when you've finished creating the theme that you want, then you can start thinking about removing the themes that you don't want.

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

Removing a Theme

Some themes are incompatible with your company image. So you go to the themes directory and delete the directory containing the offending theme.

Next time the browser is started the theme will be absent – and don't worry, if someone was using the theme that you've deleted then they move to the next theme on the list by default in their next session.

Summary

Chapter 1 has been a nice, gentle introduction into the world of customizing SugarCRM. In the course of the chapter you learned how to customize the look and feel of the SugarCRM screen – without affecting any of the functionality.

We have changed the SugarCRM URL to something that relates to the project. Installation of the tool will give a name to the directory by default; this can be changed for customization. However, do not forget the directory name is the main part of the SugarCRM URL.

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary

Customizing tab names will make them more specific catering to the need of the company. You can also get creative with the browser title. Also you can change the text on the screen to appropriate terminology to suit the company requirements. You can play around with SugarCRM themes and add the company logo.

After all this, you should be able to produce a SugarCRM implementation that has the appropriate look and feel for the project that you're working on – one that would make Pygoscelsis proud.

In Chapter 2 we look at how we can start customizing the application content, so that you are able to give each user exactly what they need – at their fingertips.

e207e1f6d17cc5a62b77c0097ebf0a7c
ebruary