Quick answers to common problems

Salesforce CRM Admin Cookbook

Over 40 recipes to make effective use of Salesforce CRM with the use of hidden features, advanced user interface techniques, and real-world solutions

Paul Goodey

[PACKT] PUBLISHING enterprise
professional expertise distilled

Salesforce CRM Admin Cookbook

Over 40 recipes to make effective use of Salesforce CRM with the use of hidden features, advanced user interface techniques, and real-world solutions

Paul Goodey

[PACKT] PUBLISHING enterprise
professional expertise distilled

BIRMINGHAM - MUMBAI

Salesforce CRM Admin Cookbook

First published: March 2013

Production Reference: 1180313

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84968-424-8

www.packtpub.com

Cover Image by Artie Ng (artherng@yahoo.com.au)

Credits

Author
Paul Goodey

Reviewers
Keir Bowden
Veturi JV Subramanyeswari

Acquisition Editor
Pramila Balan

Lead Technical Editor
Azharuddin Sheikh

Technical Editors
Vrinda Amberkar
Jalasha D'costa

Project Coordinator
Leena Purkait

Proofreaders
Stephen Silk
Bernadette Watkins

Indexer
Monica Ajmera Mehta

Production Coordinators
Aparna Bhagat
Arvindkumar Gupta

Cover Work
Aparna Bhagat

About the Author

Paul Goodey is the author of *Salesforce CRM: The Definitive Admin Handbook* by *Packt Publishing*.

He has over 15 years of experience in developing web technology solutions for companies of all sizes across a variety of industries, and has been building solutions with Salesforce CRM since 2006.

Paul has enjoyed a variety of roles while working with Salesforce CRM, having worked as a System Administrator, Developer, Business Analyst and Consultant to provide solutions for both in-house and consultancy-based end users.

Based in the UK, near London, Paul's professional qualifications include Salesforce.com Certified Developer DEV-401. In his spare time, he is a keen runner, having run several marathons and half-marathons since 2001.

Paul is a keen and active member of the salesforce.com administrator and developer online community and can be found at LinkedIn at `http://www.linkedin.com/in/paulgoodey`.

Acknowledgement

Writing this book has been fun and I have enjoyed the time I have spent working on the chapters. My family has been very supportive, as well as being a useful springboard for some of the ideas; it is only with their support and patience that I have been able to complete my part of this endeavor.

Many other people have helped in the creation of the book and I am grateful to them all. Here's my attempt to provide acknowledgment where it is rightly due.

First I'd like to thank the heroes at salesforce.com for providing such an amazing product. Salesforce CRM is one of the very few business applications that is so easy and fun to work with. The application just goes from strength to strength with each new release and the amount of innovation, new feature development, and added business value that it brings is simply outstanding.

Next I would very much like to thank the team at Packt Publishing who have successfully orchestrated the completed work. From the time when I was first approached, to the time of editing and coordinating the final reviews, the team has been extremely supportive and highly professional throughout the writing process.

I would like to thank all the many people who have contributed with ideas and code recipes. Instead of trying to name them all and risk not mentioning others, I would like to thank in general all the salesforce.com employees, professionals, and keen enthusiasts who help make the salesforce.com community such a productive and collaborative environment.

If you haven't participated in the salesforce.com online user communities such as: `success.salesforce.com` (where you can post questions or ideas); `developer.force.com`; LinkedIn salesforce.com user groups; and Twitter (look out for `#salesforce` and `#askforce`), I would strongly recommend them as they are a truly valuable place to exchange information.

Finally, I would like to thank YOU for purchasing the book. I sincerely hope you find it as enjoyable and useful to read as it has been to write!

About the Reviewers

A 25-year veteran of the IT industry, **Keir Bowden** has been working with Salesforce and the Force.com platform since late 2008 and holds all Salesforce.com certifications from Administrator through to Technical Architect. He is also a two-time Force.com MVP, frequent blogger on Apex and Visualforce solutions, and speaker at events such as Dreamforce and Cloudstock.

Keir is the Chief Technical Officer of BrightGen, a Salesforce.com Platinum Cloud Alliance Partner in the United Kingdom.

Sree (aka **Veturi JV Subramanyeswari**) is currently working as a Solution Architect at a well-known software consulting MNC in India. Prior to joining this company, she served few Indian MNCs, many startups, and R&D sectors in various roles such as programmer, tech lead, research assistant, architect, and so on. She has more than eight years of experience in web technologies covering media and entertainment, publishing, healthcare, enterprise architecture, manufacturing, public sector, defense communication, and gaming. She is also a well-known speaker who delivers talks on Drupal, Open Source, PHP, and Women in Technology, among other topics.

Sree has reviewed other tech books including:

- DevOps
- Twitter Bootstrap
- Drupal 7 Multi Sites Configuration
- Building Powerful and Robust Websites with Drupal 6
- Drupal 6 Module development
- PHP Team Development
- Drupal-6-site-blueprints
- Drupal 6 Attachment Views
- Drupal E-Commerce with Ubercart 2.x
- Drupal 7: First Look
- Twitter bootstrap
- Drupal SEO
- and many more

I would like to thank my family and friends who supported me in completing my reviews on time with good quality.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

PACKTLIB™

http://PacktLib.PacktPub.com

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read, and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Instant updates on new Packt books

Get notified! Find out when new books are published by following @PacktEnterprise on Twitter, or the *Packt Enterprise* Facebook page.

Table of Contents

Preface 1

Chapter 1: Working with Home Page Components and Custom Links 5

Introduction 5
Creating a Personal Setup link using the standard Custom Links on the sidebar 7
Using Custom Links to open Training in a new window from the sidebar 11
Creating a news-ticker message on the home page 16
Automatically collapsing Chatter feeds on the home page 20
Removing Chatter feeds on the home page 25
Adding a Send An Email button on the home page 29
Showing Opportunity Sales Stages descriptions on the home page 34

Chapter 2: Advanced User Interface 43

Introduction 43
Displaying Case Priority flags using a formula field and salesforce.com images 44
Building an Account Revenue indicator using a formula field and custom images 46
Creating a clutter-free Account tab using Visualforce 54
Showing a tabbed Account Detail page using Visualforce 57
Rendering an Account credit score graphically using JavaScript, CSS, 62
and Visualforce 62
Presenting an Account credit score graphically using a Google image chart 66

Chapter 3: Exposing Hacks and Hidden Features 75

Introduction 75
Extending the year range to 2025 on calendar pop ups for all date fields in Salesforce 76

Populating a historic year range from 1920 on calendar pop ups for all date fields in Salesforce 82
Creating a custom Log A Call button from accounts with prepopulated fields 89
Presenting an integrated view of cases on the home page 94
Removing the "discover what's new in this release" button from the home page 99

Chapter 4: Automating Salesforce CRM 105
Introduction 105
Deriving year and month values from the opportunity close date using a formula 106
Calculating the week number for the opportunity close date using a formula 108
Creating an opportunity close date e-mail alert using workflow 111
Setting a default opportunity name using a field update workflow 122
Generating a default opportunity close date using a field update workflow 129

Chapter 5: Improving Data Quality in Salesforce CRM 137
Introduction 137
Stopping non-system administrators from changing account names with a validation rule 138
Enforcing the use of two-letter account country codes using a validation rule 140
Validating if US zip codes are well formed using a validation rule with REGEX 143
Validating if UK postcodes are well formed using a validation rule with REGEX 145
Creating a mandatory Reason Lost field for lost opportunities using a validation rule 148

Chapter 6: Implementing Approval Processes 153
Introduction 153
Creating an e-mail template for use with approval assignments 154
Setting up a user to be associated with an approval process 156
Building an opportunity approval process for deals greater than USD 100,000 158

Chapter 7: Productivity Tools for Superusers and Advanced Administration 173
Introduction 173
Converting a 15-digit Salesforce opportunity ID to 18 digits using a formula field 174
Generating help pages for custom objects using Visualforce and a PDF document 177
Building a mass delete opportunity button using a custom list button 180
Chapter 8: Configuring and Installing Salesforce for Outlook E-mail Integration 189
Introduction 189
Enabling the Email to Salesforce feature in Salesforce CRM 190
Obtaining system requirements information for Salesforce for Outlook 194
Setting up Salesforce for Outlook configurations within Salesforce CRM 196
Installing the Salesforce for Outlook software on a local machine 204
Configuring the Salesforce for Outlook software on a local machine 213
Chapter 9: Integrating Salesforce CRM with External Online Tools 223
Introduction 223
Providing a Google News search for an account using a custom link 224
Building a custom Web Tab to display an external web application 228
Displaying the location of an organization using a Google Map and a Visualforce page 234
Index 245

Preface

As an industry-leading customer relationship management (CRM) application, Salesforce CRM helps the enterprise to improve client relations. It greatly enhances sales performance and provides your business with a robust CRM system.

In order to achieve optimum performance and benefits from Salesforce CRM, there are many functional areas and techniques for you, as the Salesforce Administrator, to consider. This is the only cookbook-style publication that provides recipes which focus specifically on configuring and extending this powerful CRM system.

Salesforce CRM Admin Cookbook provides step-by-step instructions that enable you to instantly extend and unleash the power of Salesforce CRM. This practical cookbook contains recipes that are presented using clear and comprehensive instructions along with detailed screenshots, code, and comments.

Whether you are looking for solutions to enhance the core features or are looking for ideas for advanced customization techniques, this book will provide you with immediate, practical, and exciting real-world recipes.

What this book covers

Chapter 1, *Working with Home Page Components and Custom Links*, covers recipes that enhance and extend the features and functionality associated with the Salesforce home page by using Home Page Components along with HTML, CSS, and JavaScript code.

Chapter 2, *Advanced User Interface*, includes advanced techniques and solutions that can be used to enhance and override the native Salesforce user interface and add graphical elements by using images, HTML, CSS, JavaScript, and Visualforce.

Chapter 3, *Exposing Hacks and Hidden Features*, introduces recipes that modify the behavior of standard functionality and elements of Salesforce CRM (such as extending the year range selection on the calendar pop-up control) using HTML and JavaScript.

Chapter 4, *Automating Salesforce CRM*, provides standard Salesforce mechanisms that automate the Salesforce CRM system and helps to deliver best practice data management using Salesforce workflow and formulas.

Chapter 5, *Improving Data Quality in Salesforce CRM*, covers recipes that provide record validation and improve data quality using advanced Salesforce CRM validation rule mechanisms.

Chapter 6, *Implementing Approval Processes*, uses step-by-step instructions to describe the configuration of a complete approval process using the standard Salesforce CRM approval mechanism.

Chapter 7, *Productivity Tools for Superusers and Advanced Administration*, offers techniques and solutions to improve admin productivity (for example, when mass-deleting records) by using custom buttons, formulas, and Visualforce.

Chapter 8, *Configuring and Installing Salesforce for Outlook E-mail Integration*, provides complete step-by-step instructions for the configuration and installation of Outlook e-mail integration with the Salesforce CRM application.

Chapter 9, *Integrating Salesforce CRM with External Online Tools*, contains advanced recipes that add data and functionality from external online systems that are served within Salesforce CRM using HTML, CSS, JavaScript, and Visualforce.

What you need for this book

The pre-requisite for this book is a computer with an Internet connection with one of the following supported browsers: Microsoft Internet Explorer, Mozilla Firefox, Google Chrome or Apple Safari. You need either an Enterprise, Unlimited, or a Developer edition of Salesforce CRM along with System Administrator permission.

Who this book is for

This book is for Salesforce administrators and developers who want to quickly incorporate enhanced functionality and extend the power of Salesforce CRM. Whether you are a Salesforce novice or a more experienced admin, this book provides practical, step-by-step instructions in the use of hidden features, advanced user interface techniques, and solutions for process automation, plus data and systems integration.

Not only are standard Salesforce CRM features covered, such as workflow and approval processes, validation rules, and formula fields, but you will also be exposed to additional technologies including HTML, JavaScript, CSS, Apex, and Visualforce.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "Within this component we are using the `<iframe>` HTML tag which allows us to insert the rendered output of a specified Visualforce page."

A block of code is set as follows:

```
<center>
<div style="border: 1px solid rgb(51, 153, 255); width: 96%; color:
black; font-size: 18px; background-color: rgb(255, 255, 204);"
id="NewsTicker">
<marquee onmouseover="this.scrollAmount=0" onmouseout="this.
scrollAmount=6" width="98%">Welcome to the Salesforce CRM Admin
cookbook</marquee></div></center><br>
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes, for example, appear in the text like this: "Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**".

> Warnings or important notes appear in a box like this.

> Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Working with Home Page Components and Custom Links

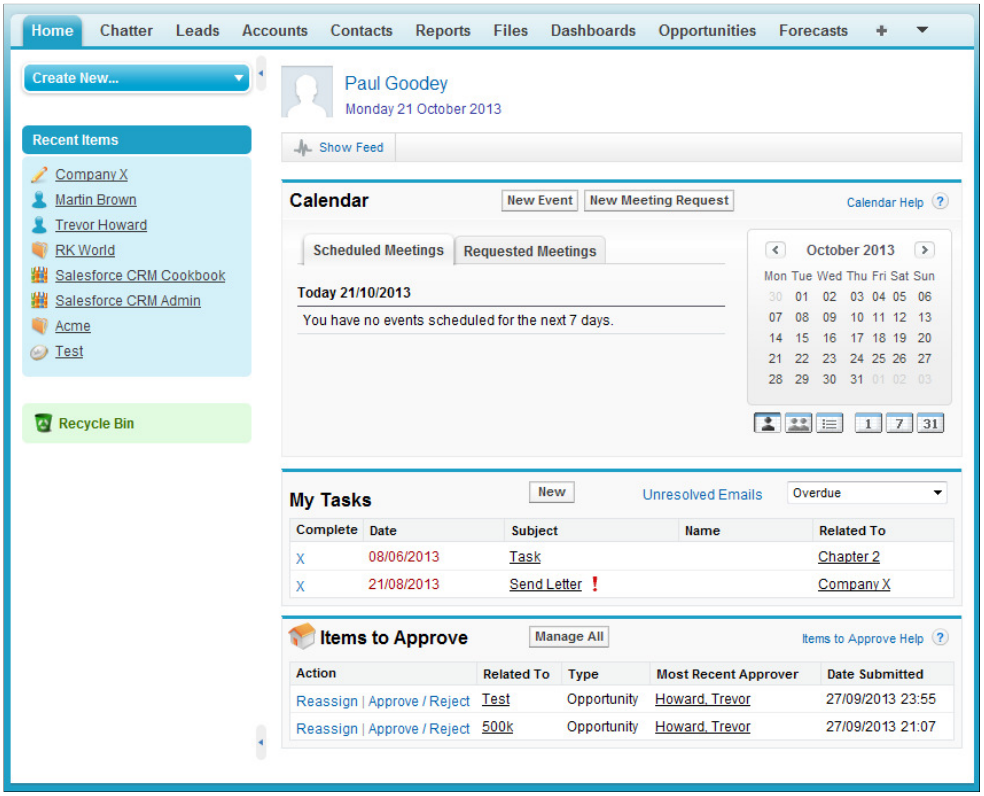
In this chapter, we will cover the following recipes:

- Creating a Personal Setup link using the standard Custom Links on the sidebar
- Using Custom Links to open Training in a new window from the sidebar
- Creating a news-ticker message on the home page
- Automatically collapsing Chatter feeds on the home page
- Removing Chatter feeds on the home page
- Adding a Send An Email button on the home page
- Showing Opportunity Sales Stage descriptions on the home page

Introduction

The **Home** tab in Salesforce CRM is generally set as the opening page for users when they first log in to the application and provides a great way for users to view Chatter posts, access their calendar, tasks, items to approve, and so on.

The following screenshot shows the Salesforce CRM **Home** screen:



The **Home** tab allows various standard home page components to be shown, such as **Calendar**, **My Tasks**, or **Items to Approve**, (as shown in the preceding screenshot), that you are able to administer and choose whether to display for the various users in your organization.

Home page components can be presented to users by customizing the home page layouts where you can assign different home page layouts to different users based on their profile.

You can also create your own custom home page components that will display alongside the standard components and you can choose to make custom sidebar components display on all pages within the application or only on the **Home** tab.

The following recipes are designed to improve the layout of the home page. The recipes also provide additional features and functionality that are not provided natively within the Salesforce CRM application.

Creating a Personal Setup link using the standard Custom Links on the sidebar

All users need to change their personal settings, from time to time, in the Salesforce CRM application. They may, for example, wish to edit their user information, change their password, or you may need them to grant login access to administrators, plus many other reasons.

Accessing the **Personal Setup** area is done by users clicking on their name, looking for the **Setup** link in the drop-down list, clicking on the **Setup** link, and then finally clicking on the **Personal Setup** link in the sidebar.

All this takes time and can often be a challenge for less-experienced users of the application. By providing a direct shortcut link in the sidebar, all users will be able to access their **Personal Setup** area with a single click, and save their time and efforts.

How to do it...

Carry out the following steps to create a **Personal Setup** link in the sidebar:

1. Navigate to the home page components' setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.

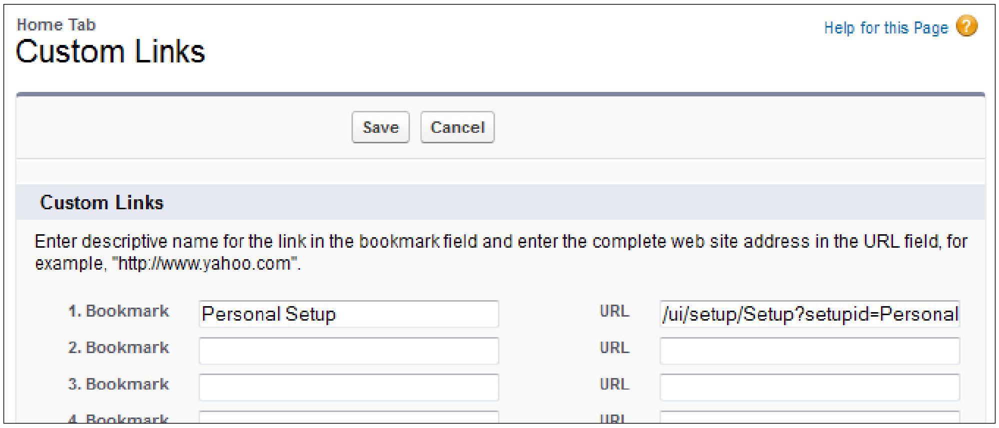
   > Locate the **Custom Links** row within the **Standard Components** section.

2. Click on **Edit**.

   > Within the **Custom Links** page you can enter a maximum of 15 links.

3. Enter `Personal Setup` in the **1. Bookmark** field.

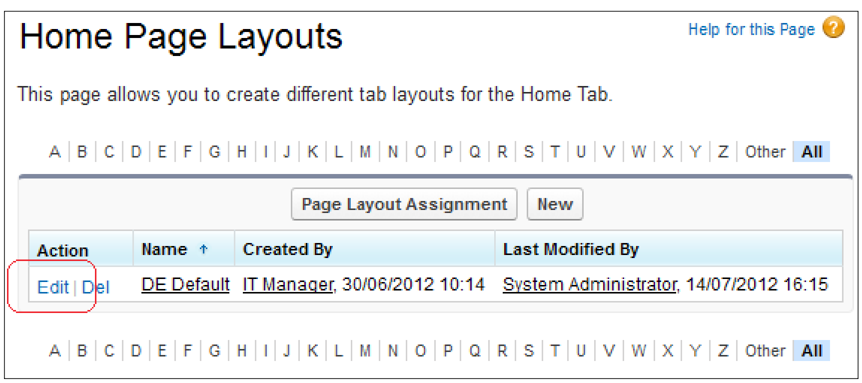
4. Enter `/ui/setup/Setup?setupid=PersonalSetup` in the corresponding **URL** field, as shown in the following screenshot:



5. Click on **Save**.

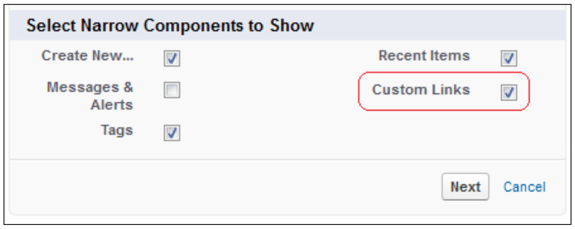
> We now need to add the standard Custom Links component to a home page layout (if it has not been already added).

6. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
7. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**, as shown in the following screenshot:

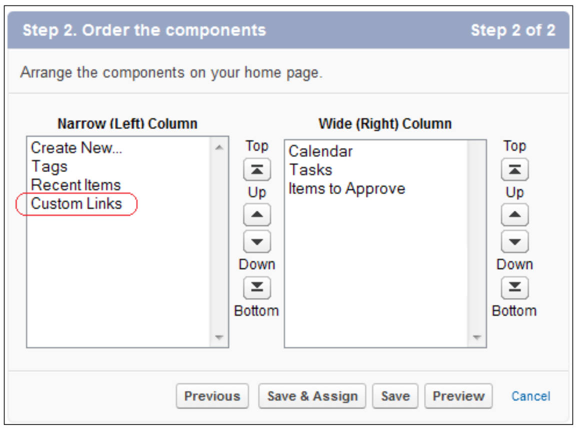


We will be presented with the **Step 1. Select the components to show** page.

8. Check the **Custom Links** checkbox in the **Select Narrow Components to Show** section, as shown in the following screenshot:



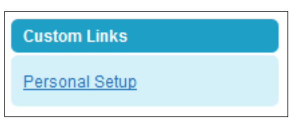
9. Click on **Next**.
10. Move **Custom Links** to the top position in the **Narrow (Left) Column** using the **Arrange the component on your home page** section, as shown in the following screenshot:



11. Click on **Save**.

How it works...

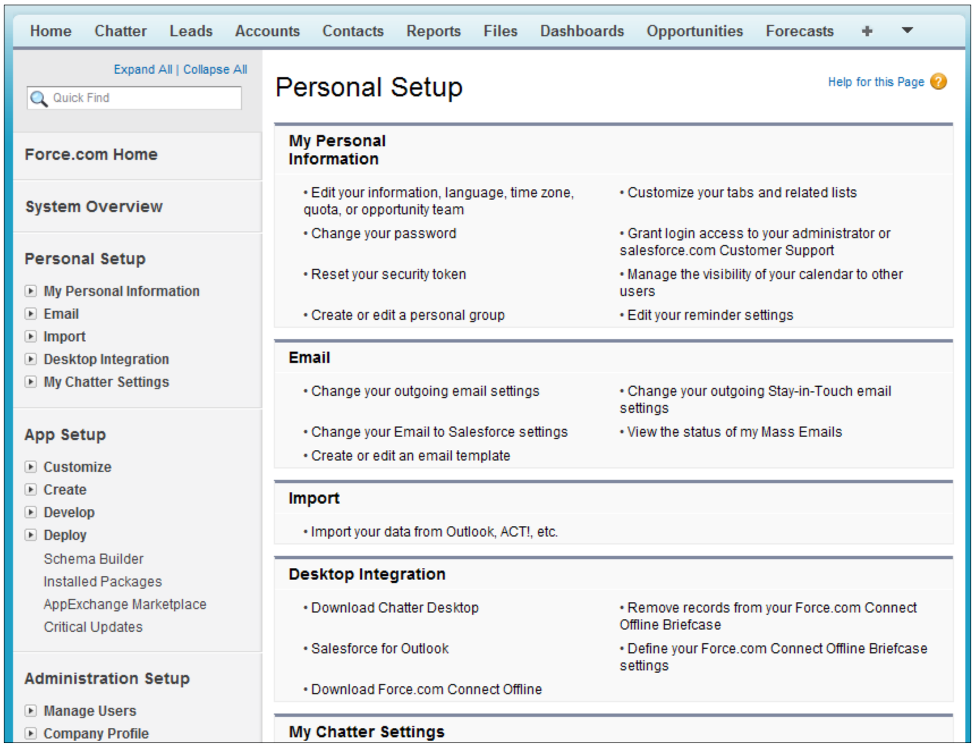
The link appears in the sidebar within the standard **Custom Links** section, as shown in the following screenshot:



When the link is clicked, the user is immediately presented with their **Personal Setup** page.

There's more...

Clicking on the link displays the **Personal Setup** page in the same window and is useful when there is no requirement for the link to open up in a new browser window. The following screenshot shows the result of clicking on the **Personal Setup Custom Link**:



See also

The *Using Custom Links to open Training in a new window from the sidebar* recipe in this chapter.

Using Custom Links to open Training in a new window from the sidebar

In the Salesforce CRM application, there are various options for help and training.

Accessing the training area is done by the users by clicking on the **Help** link at the top of the page (which then opens in a new browser window). Users then need to look for the **Training** tab within the new page and then click on the tab.

All this takes a little time and can often be a challenge for less-experienced users of the application. By providing a direct shortcut link in the sidebar, all users will be able to open **Training** automatically in a new window with a single click, thus saving time and effort.

How to do it...

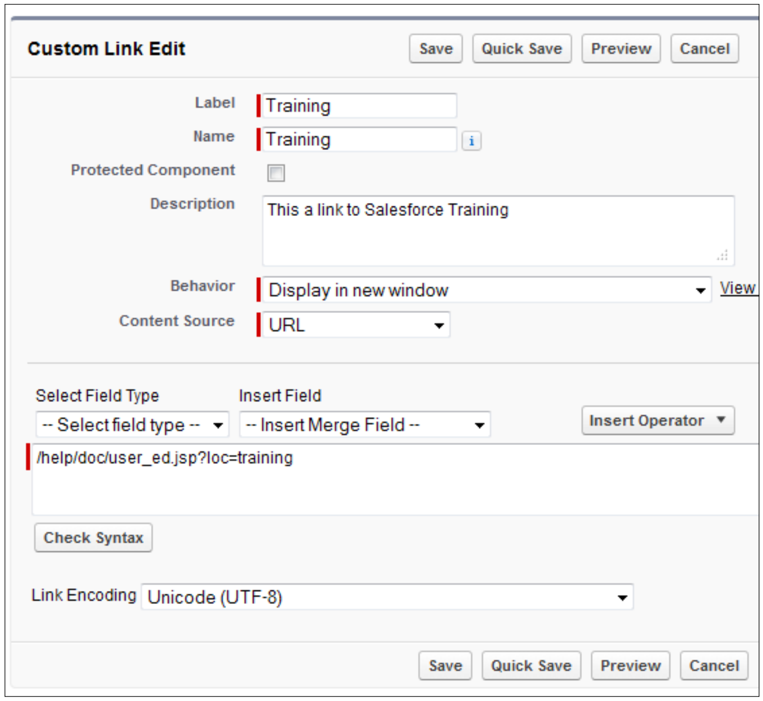
Carry out the following steps to create a link in the sidebar to open **Training** in a new window:

1. Navigate to the Custom Links home page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Custom Links**.
2. Click on **New**.
3. Enter the label of the Custom Link in the **Label** field. Here, type the text **Training**.
4. Accept the default name of the Custom Link in the **Name** field, **Training**.
5. Leave the **Protected Component** checkbox unchecked.

> The **Protected Component** option is used by developers to mark the Custom Link as protected in managed packages. This then allows the developer to delete the link in any future releases of the managed package without worrying about causing package installations to fail.

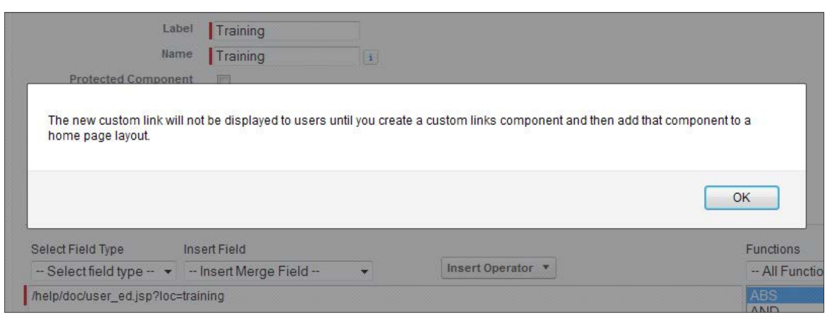
6. Enter the following description in the **Description** field: `This a link to Salesforce Training`.
7. Choose the **Display in new window** option from the **Behavior** picklist. Choose the **URL** option from the **Content Source** picklist.

8. Enter `/help/doc/user_ed.jsp?loc=training` into the source section as shown in the following screenshot:



9. Ensure the selection **Unicode (UTF-8)** is set in the **Link Encoding** picklist.
10. Click on **Save**.

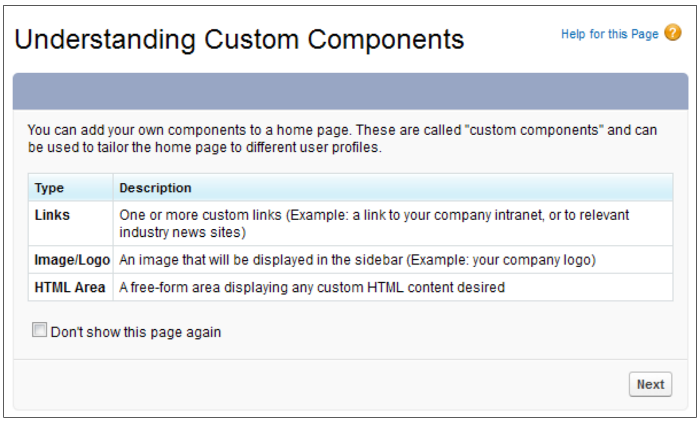
    We now need to create a custom home page component to house this custom link. The alert displayed in the following screenshot reminds us of that:



11. Click on **OK**.

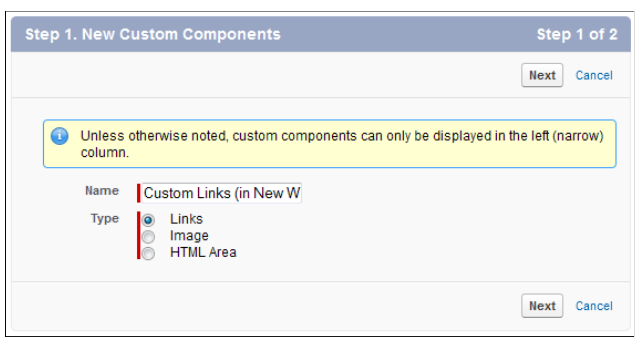
12. Now navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
13. Click on **New**.
14. Click on **Next** (on the **Understanding Custom Components** splash screen, if shown).

    The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page again** checkbox has not previously been checked) as in the following screenshot:



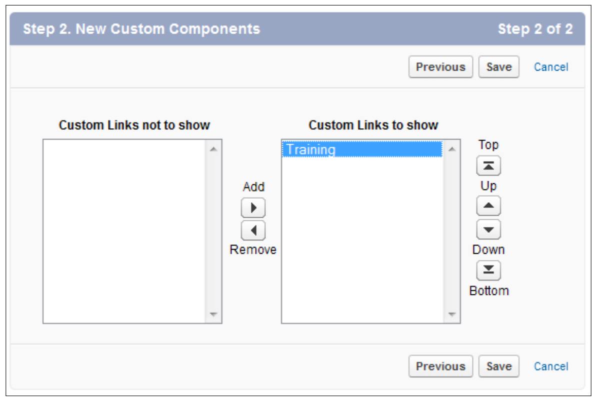
    Here, we will be presented with the **Step 1. New Custom Components** page.

15. Enter the name of the **Custom Component** in the **Name** field. Enter the text `Custom Links (in New Window).`
16. Select the **Links** option from the **Type** options list as shown in the following screenshot:



17. Click on **Next**.

18. Now add the **Training** link to the list of **Custom Links to show** as shown in the following screenshot:

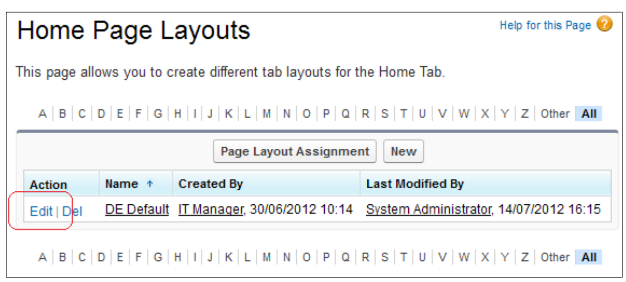


19. Click on **Save**

> We have created our **Training** link's custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

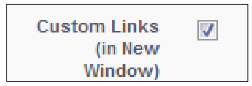
20. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.

21. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**, as shown in the following screenshot:

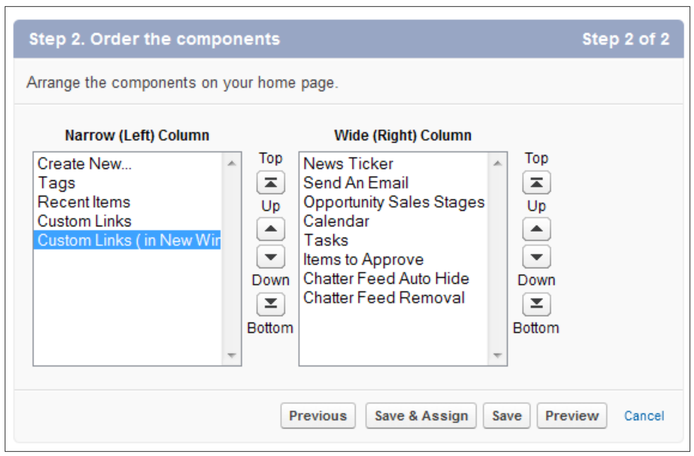


We will be presented with the **Step 1. Select the components to show** page.

22. Check the **Custom Links (in New Window)** checkbox in the **Select Narrow Components to Show** section as shown in the following screenshot:



23. Click on **Next.**
24. Move **Custom Links (in New Window)** to the top position in **Narrow (Left) Column** using the **Arrange the component on your home page.** section, as shown in the following screenshot:



25. Click on **Save**.

How it works...

Clicking on the **Training** link opens a new smaller browser window with the Salesforce **Training** page directly accessed and loaded alongside the main Salesforce CRM application windows. Users can switch back to the main application when they want and simply close the **Training** window when they are finished viewing it.

You can see what this looks like in the following screenshot:

See also

The *Creating a Personal Setup link using the standard Custom Links on the sidebar* recipe in this chapter.

Creating a news-ticker message on the home page

Animated text messages and custom HTML styling can be very effective in gaining users' attention.

Not only do they allow a reasonably large number of characters to be presented on a single line of text but they also provide a visually dynamic "headline news" style of message delivery for what can otherwise be overlooked static text.

You can draw attention to your news message on the Salesforce home page using this custom home page component recipe with the steps listed in the following section.

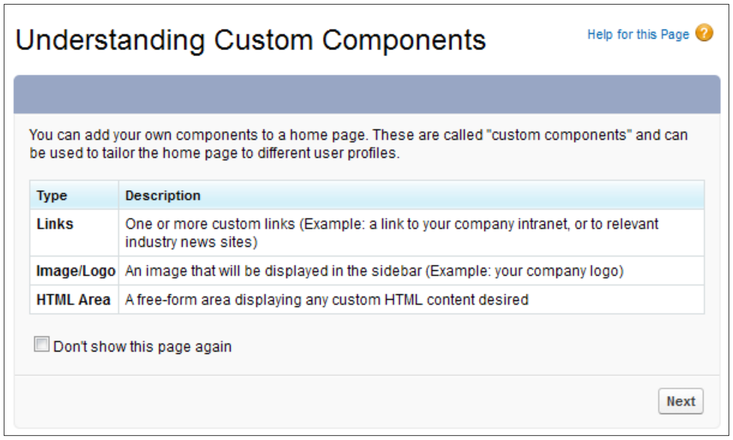
How to do it...

Carry out the following steps to create a news-ticker message on the home page:

1. Navigate to the home page components setup page, by clicking the following: **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

   > The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown). The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page again** checkbox has not previously been checked) as in the following screenshot:

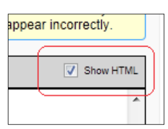


Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the custom component in the **Name** field. Enter the text `News Ticker`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure that the **Wide (Right) Column** option is selected within the **Component Position** option list.

> You are unable to change this setting after the component is created.

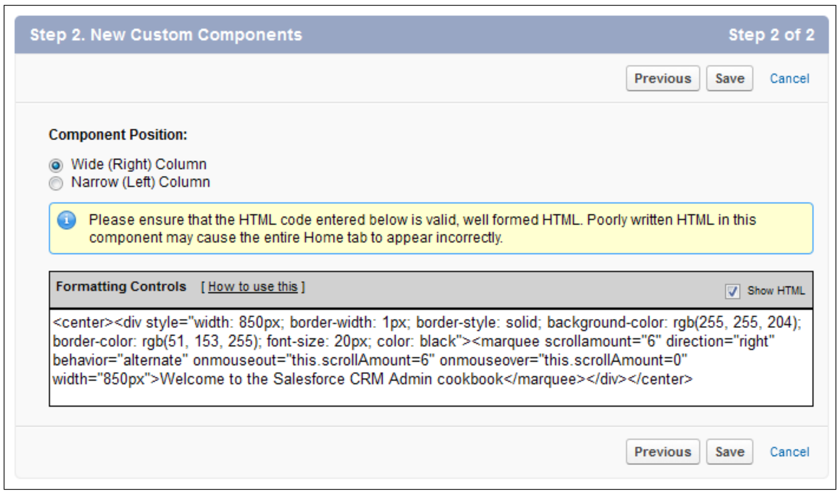
8. Check the **Show HTML** checkbox.

> The above step is important! Locate and check the **Show HTML** checkbox, as shown in the following screenshot:
>
> 
>

9. Paste the following code (as shown in the following screenshot):

```
<center>
<div style="border: 1px solid rgb(51, 153, 255); width: 96%;
color: black; font-size: 18px; background-color: rgb(255, 255,
204);" id="NewsTicker">
```

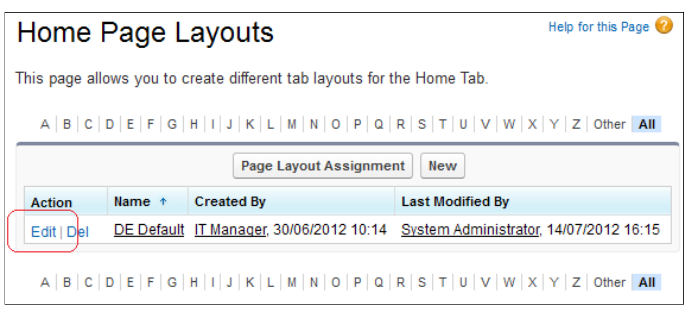
```
<marquee onmouseover="this.scrollAmount=0" onmouseout="this.
scrollAmount=6" width="98%">Welcome to the Salesforce CRM Admin
cookbook</marquee></div></center><br>
```



10. Click on **Save**.

> We have created our **News Ticker** custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

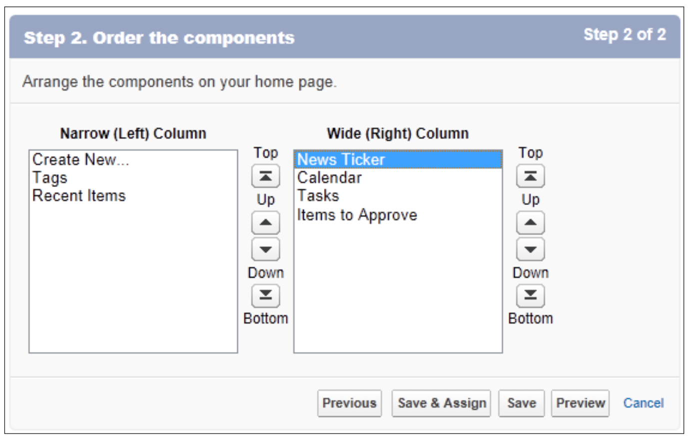
11. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**, as shown in the following screenshot:



We will be presented with the **Step 1. Select the components to show** page.

13. Check the **News Ticker** checkbox in the **Select Wide Components to Show** section, as shown in the following screenshot:

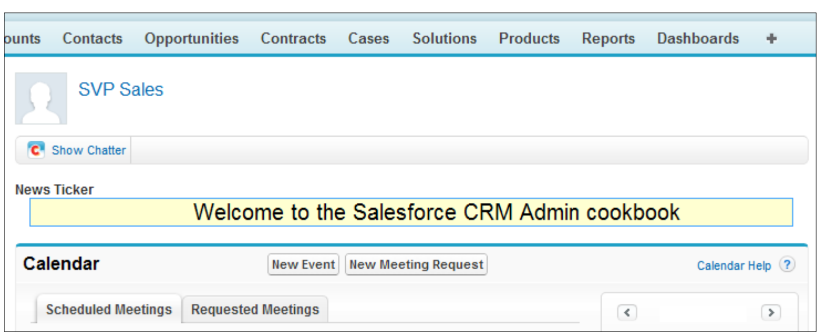
14. Click on **Next**.
15. Move **News Ticker** to the top position in **Wide (Right) Column** using the **Arrange the component on your home page.** section, as shown in the following screenshot:



16. Click on **Save**.

How it works...

By using HTML and in particular the HTML marquee tag and **Cascading Style Sheets** (**CSS**), we are able to present a moving text section. You can see what this looks like in the following screenshot:



There's more...

Replace the text `Welcome to the Salesforce CRM Admin cookbook` with a suitable message of your choice.

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard) you must ensure that the code is valid.

> Pay particular attention to the displayed warning message:
>
> **Please ensure that the HTML code entered below is valid, well formed HTML. Poorly written HTML in this component may cause the entire Home tab to appear incorrectly**

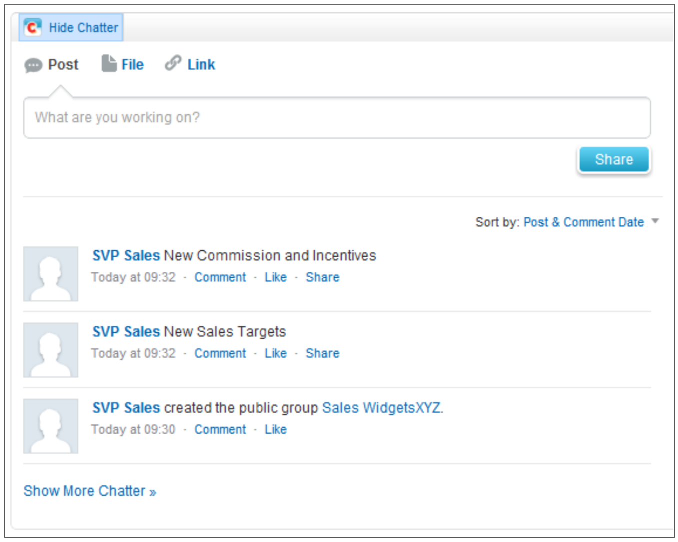
There is a maximum of 20 custom components that can be added to a Home Page layout.

> The marquee tag is a non-standard HTML element which causes text to scroll up, down, left or right automatically. From `http://en.wikipedia.org/wiki/Marquee_element`:
>
> *The marquee element was first invented for Microsoft's Internet Explorer and is still supported by it. Firefox, Opera, Chrome, and Safari web browsers support it for compatibility with legacy pages.*

Automatically collapsing Chatter feeds on the home page

The **Chatter** feed on the home page appears at the top of the home page layout (as shown in the screenshot in the *Introduction* section) and is a great place for presenting the feed of **Chatter** posts.

However, since the **Chatter** feed is always shown at the top of the home page, users find themselves having to scroll down the home page to get to their calendar, items to approve, or anything else that has been configured on their home page, as shown in the following screenshot:



The **Hide Chatter** button allows users to manually hide the feed but this has to be done every time the user lands on the home page either when they first log in or when they click on the **Home** tab.

To save users' time by automatically hiding chatter feeds on the Home Page every time the user accesses the **Home** tab create this custom Home Page Component using the steps listed in the following section.

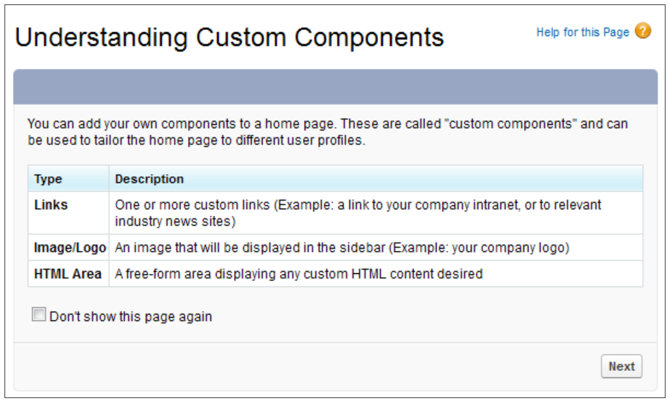
How to do it...

Carry out the following steps to automatically collapse **Home Page Chatter** feeds:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown). The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page again** checkbox has not previously been checked), as shown in the following screenshot:

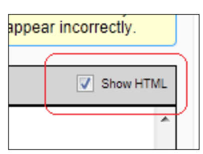


Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the custom component in the **Name** field. In this recipe enter the text, `Chatter Feed Auto Hide`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure that the **Wide (Right) Column** option is selected within the **Component Position** option list.

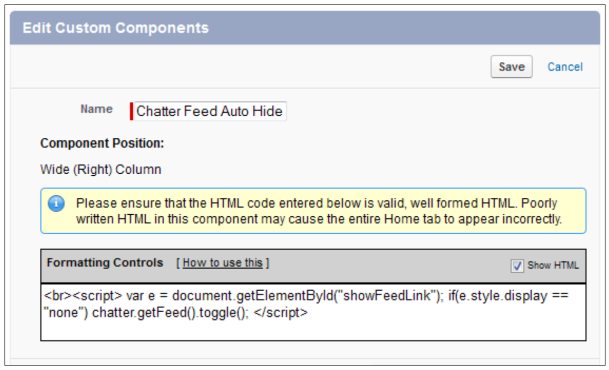
> You are unable to change this setting after the component is created.

8. Check the **Show HTML** checkbox.

> The above step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:
>
> 
>

9. Paste the following code (as shown in the following screenshot):

```
<br><script> var e = document.getElementById("showFeedLink");
if(e.style.display == "none") chatter.getFeed().toggle(); </
script>
```



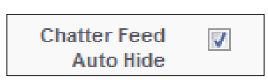
10. Click on **Save**.

> We have created our **Chatter Feed Auto Hide** custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

11. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**.

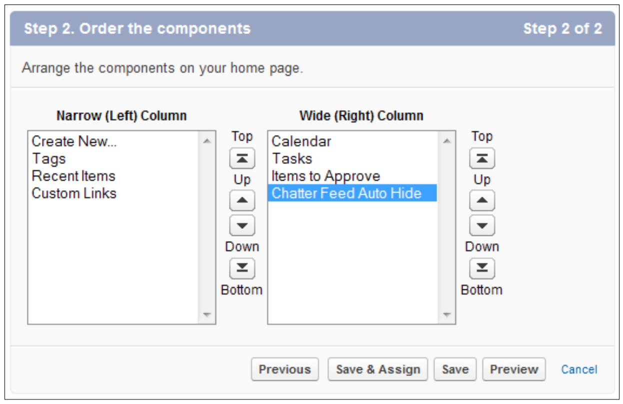
    We will be presented with the **Step 1. Select the components** to show page.

13. Check the **Chatter Feed Auto Hide** checkbox in the **Select Wide Components to Show** section, as shown in the following screenshot:



14. Click on **Next.**

15. Move the **Chatter Feed Auto Hide** to the top position in **Wide (Right) Column** using the **Arrange the component on your home page** section, as shown in the following screenshot:



16. Click on **Save**.

There's more...

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard) you must ensure that the code is valid.

> Pay particular attention to the displayed warning message:
>
> **Please ensure that the HTML code entered below is valid, well formed HTML. Poorly written HTML in this component may cause the entire Home tab to appear incorrectly**

There is a maximum of 20 custom components that can be added to a home page layout.

> This recipe introduces an element of risk and is not supported by `salesforce.com`.
>
> If `salesforce.com` chooses to rename the HTML ID, used for the **Hide Chatter** button, this recipe will cease to function. This recipe code would need to be changed to incorporate the new ID; hence this technique is not generally recommended by Salesforce.

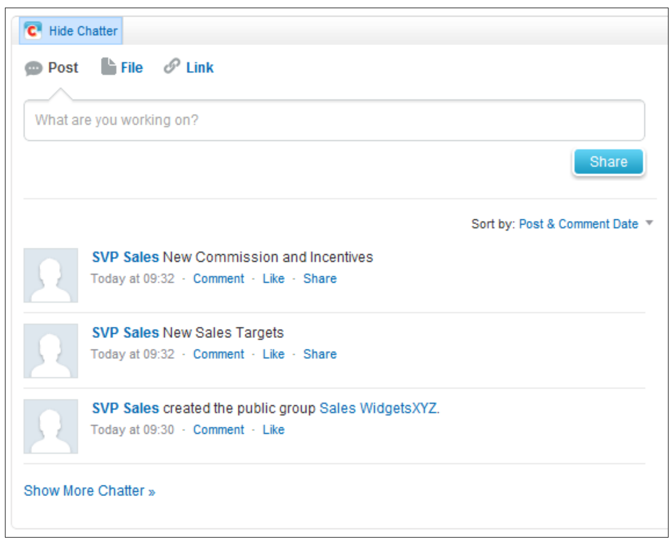
See also

The *Removing Chatter feeds on the home page recipe* in this chapter.

Removing Chatter feeds on the home page

The **Chatter** feed on the home page appears at the top of the home page layout (as shown in the screenshot in the *Introduction* section) and is a great place for presenting the feed of **Chatter** posts.

However, often there are users who are not concerned with seeing these feeds on their home page and would like to completely remove the section. This is not possible using standard `salesforce.com` configuration.



In this scenario it is a requirement to fully remove the **Chatter** section and to remove the **Hide Chatter** button as well:

You can remove **Chatter** feeds as well as the **Chatter Hide/Show** button on the home page using this custom home page component recipe with the set of steps listed in the following section.

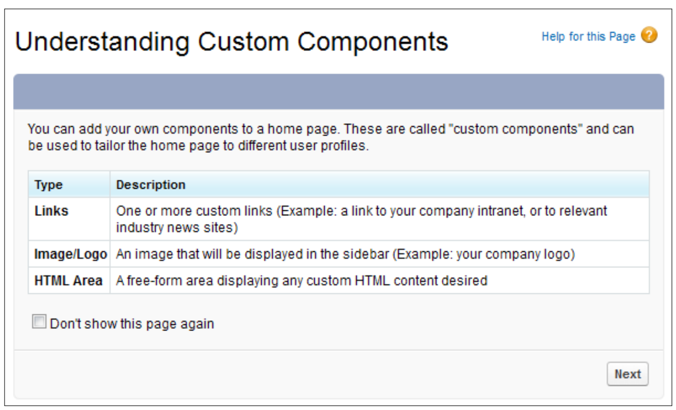
How to do it...

Carry out the following steps to remove **Chatter** feeds from the home page:

1. Navigate to the home page components setup page, by clicking the following: **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown). The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page** checkbox again has not previously been checked) as in the following screenshot:



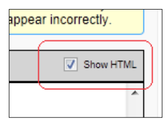
> Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the custom component in the Name field. In this recipe, enter the text `Chatter Feed Removal`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.

7. Ensure the **Wide (Right) Column** option is selected within the **Component Position** option list.

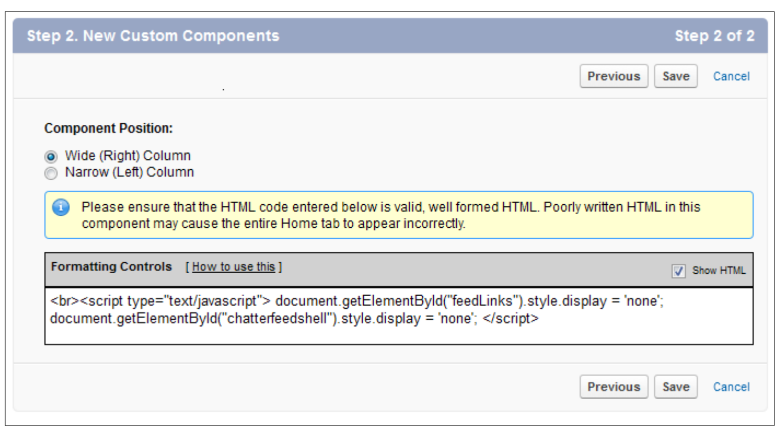
> You are unable to change this setting after the component is created.

8. Check the **Show HTML** checkbox.

> The above step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:
>
> 
>

9. Paste the following code (as shown in the following screenshot):

```
<script type="text/javascript">
document.getElementById("feedLinks").style.display = 'none';
document.getElementById("chatterfeedshell").style.display =
'none';
</script>
```



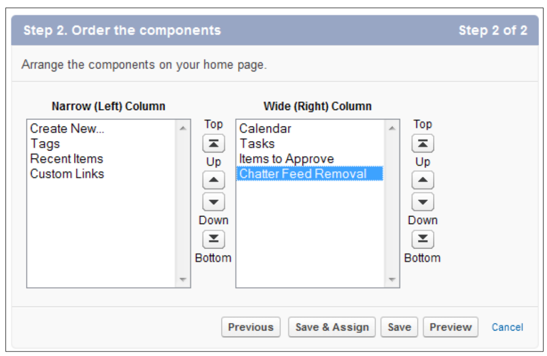
10. Click on **Save**.

> We have created our **Chatter Feed Removal** custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

11. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**.

    We will be presented with the **Step 1. Select the components to show** page.
13. Check the **Chatter Feed Removal** checkbox in the **Select Wide Components to Show** section, as shown in the following screenshot:

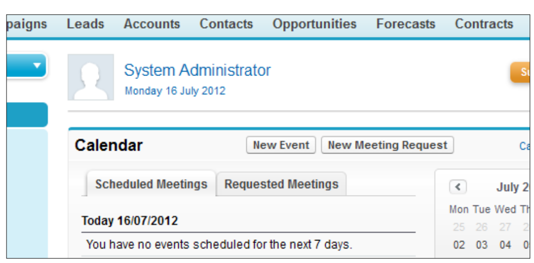
14. Click on **Next**.
15. Position the **News Ticker** as the top position in the **Wide (Right) Column** using the **Arrange the component on your home page.** section, as shown in the following screenshot:



16. Click on **Save**.

How it works...

In the following screenshot, you can see the effects of having the **Chatter** section removed from the home page:



There's more...

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard) you must ensure that the code is valid.

> Pay particular attention to the displayed warning message:
>
> **Please ensure that the HTML code entered below is valid, well formed HTML. Poorly written HTML in this component may cause the entire Home tab to appear incorrectly**

There are a maximum of 20 custom components that can be added to a Home Page layout.

> This recipe introduces an element of risk and is not supported by `salesforce.com`.
>
> If `salesforce.com` chooses to rename the HTML IDs, used for the **Chatter** section, this recipe will cease to function. This recipe's code would need to be changed to incorporate the new ID, hence this technique is not generally recommended by Salesforce.

Adding a Send An Email button on the home page

Within the Salesforce CRM application you can natively add custom links to the home page, which can be configured to call web URLs. These URLs can be either external to Salesforce or internal, such as a link to the Salesforce CRM Training pages (covered in an earlier recipe within this chapter).

Salesforce does not, however, natively provide a way of adding custom buttons onto the home page in order to call web URLs.

Often it is preferable to provide a button to invoke certain behavior (rather than a URL link). One such example is a **Send An Email** button.

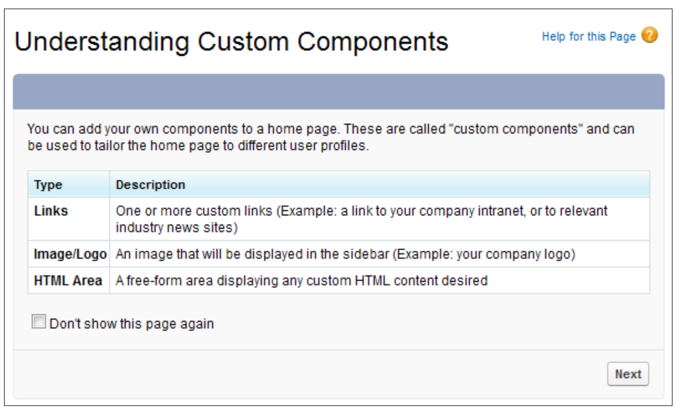
How to do it...

Carry out the following steps to add a **Send An Email** button to the home page:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

   [The **New** button is found by scrolling down the page to the **Custom Components** section.]

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown). The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page again** checkbox has not previously been checked), as shown in the following screenshot:

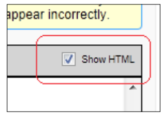
   

   Here, we will be presented with the **Step 1. New Custom Components** page.

4. Enter the name of the custom component in the **Name** field. Enter the text `Email Button`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure that the **Wide (Right) Column** option is selected within the **Component Position** option list.

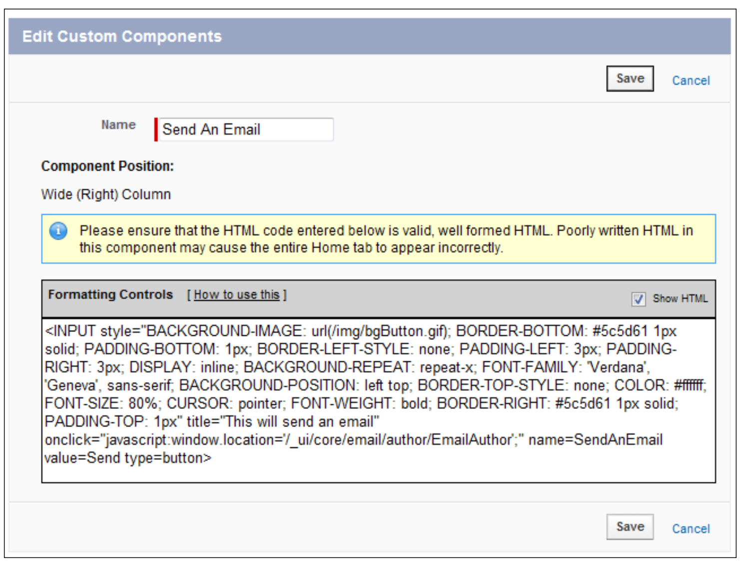
   [You are unable to change this setting after the component is created.]

8. Check the **Show HTML** checkbox.

> The above step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:
>
> 
>

9. Paste the following code (as shown in the following screenshot):

```
<INPUT style="BACKGROUND-IMAGE: url(/img/bgButton.gif); BORDER-
BOTTOM: #5c5d61 1px solid; PADDING-BOTTOM:
1px; BORDER-LEFT-STYLE: none; PADDING-LEFT: 3px; PADDING-RIGHT:
3px; DISPLAY: inline; BACKGROUND-REPEAT:
repeat-x; FONT-FAMILY: 'Verdana', 'Geneva', sans-serif;
BACKGROUND-POSITION: left top; BORDER-TOP-STYLE:
none; COLOR: #ffffff; FONT-SIZE: 80%; CURSOR: pointer; FONT-
WEIGHT: bold; BORDER-RIGHT: #5c5d61 1px solid;
PADDING-TOP: 1px" title="This will send an email"
onclick="javascript:window.location='/_ui/core/email/author/
EmailAuthor';" name=SendAnEmail value=Send
type=button>
```



10. Click on **Save**.

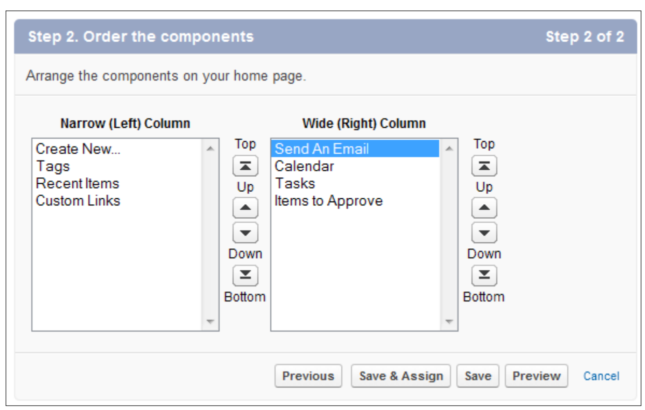
> We have created our **Send An Email** button's custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

11. Navigate to the home page components setup page, by clicking the following: **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click **Edit**. Here we are editing the home page layout named **DE Default**.

    We will be presented with the **Step 1. Select the components to show** page.

13. Check the **Send An Email** checkbox in the **Select Wide Components to Show** section as shown:

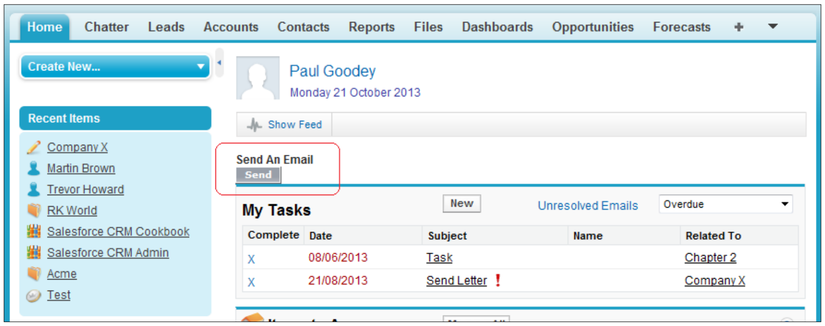
14. Click on **Next**.
15. Position the **Send An Email** button as the top position in the **Wide (Right) Column** using the **Arrange the component on your home page.** section, as shown in the following screenshot:



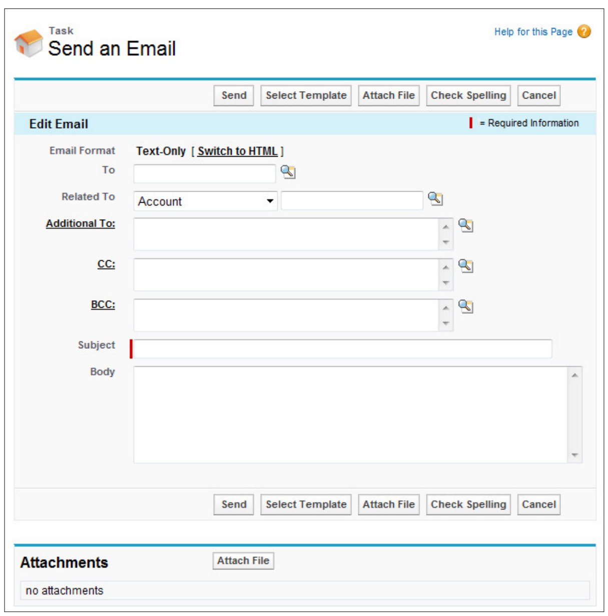
16. Click on **Save**.

How it works...

You can see what this looks like in the following screenshot:



When the button is clicked the task's e-mail page is displayed, as shown in the following screenshot:



There's more...

When entering HTML and JavaScript code into the HTML editor section (in Step 2 of the **New Custom Component** wizard) you must ensure that the code is valid.

> Pay particular attention to the displayed warning message:
>
> **Please ensure that the HTML code entered below is valid, well formed HTML. Poorly written HTML in this component may cause the entire Home tab to appear incorrectly**

There are a maximum of 20 custom components that can be added to a home page layout.

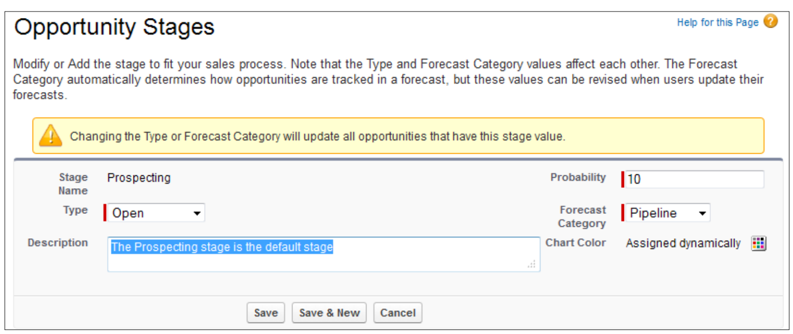
Showing Opportunity Sales Stages descriptions on the home page

The Opportunity Sales Stages configured in a Salesforce organization are important for pipeline and performance measurement. The stages should be clearly described and communicated to the sales team and across the business. In this a recipe we will describe the stages in Salesforce and make these descriptions available on the home page.

Getting ready

Carry out the following steps to add sales stages descriptions:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Fields** | **Stage**.
2. Click on **Edit** on the **Stage Name**.
3. Enter text in the **Description field**.



> Now continue to add descriptions for all your sales stages by following the steps above.

How to do it...

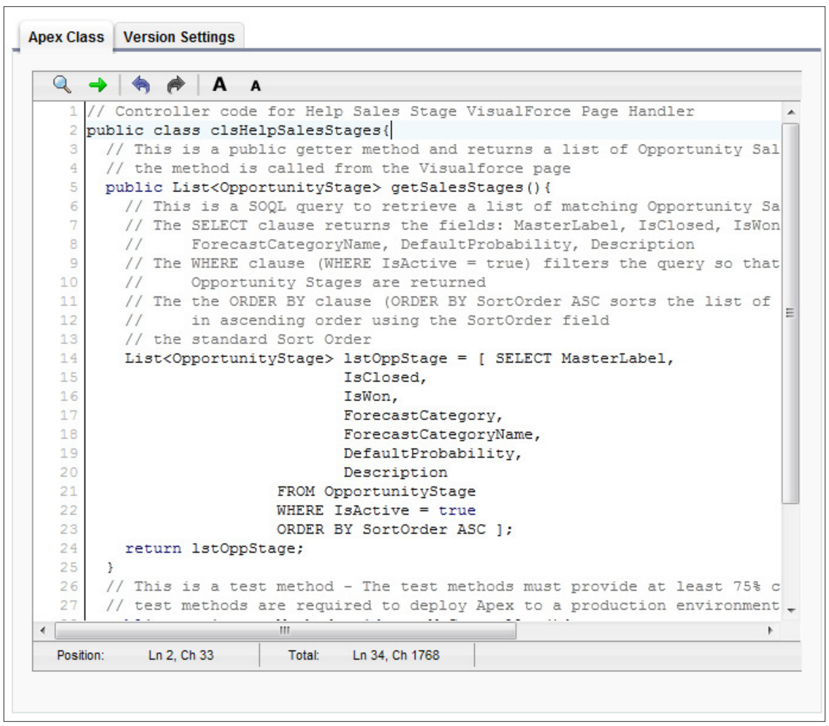
Carry out the following steps to create an Apex class to show Opportunity Sales Stages descriptions on the home page:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Develop** | **Apex Classes**.
2. Click on **New**.
3. Paste the following code (as shown in the following screenshot):

```
// Controller code for Help Sales Stage VisualForce Page Handler
public class clsHelpSalesStages{
  // This is a public getter method and returns a list of
Opportunity Sales Stages
  // the method is called from the Visualforce page
  public List<OpportunityStage> getSalesStages(){
    // This is a SOQL query to retrieve a list of matching
Opportunity Sales Stages
    // The SELECT clause returns the fields: MasterLabel,
IsClosed, IsWon, ForecastCategory,
    //     ForecastCategoryName, DefaultProbability, Description
    // The WHERE clause (WHERE IsActive = true) filters the query
so that only Active
    //     Opportunity Stages are returned
    // The the ORDER BY clause (ORDER BY SortOrder ASC sorts the
list of Opportunity Stages
    //     in ascending order using the SortOrder field
    // the standard Sort Order
    List<OpportunityStage> lstOppStage = [ SELECT MasterLabel,
                           IsClosed,
                           IsWon,
                           ForecastCategory,
                           ForecastCategoryName,
                           DefaultProbability,
                           Description
                    FROM OpportunityStage
                    WHERE IsActive = true
                    ORDER BY SortOrder ASC ];
    return lstOppStage;
  }
  // This is a test method - The test methods must provide at
least 75% code coverage
  // test methods are required to deploy Apex to a production
environment
```

```
    public static testMethod void testMyController(){
      clsHelpSalesStages objOppStage = new clsHelpSalesStages();
      List<OpportunityStage> lstOppStageTest = objOppStage
.getSalesStages();
      // This is an assertion to ensure that Opportunity Sales
Stages are returned.
      System.assert( lstOppStageTest.size() > 0 );
    }
}
```

4. Click on **Save**.

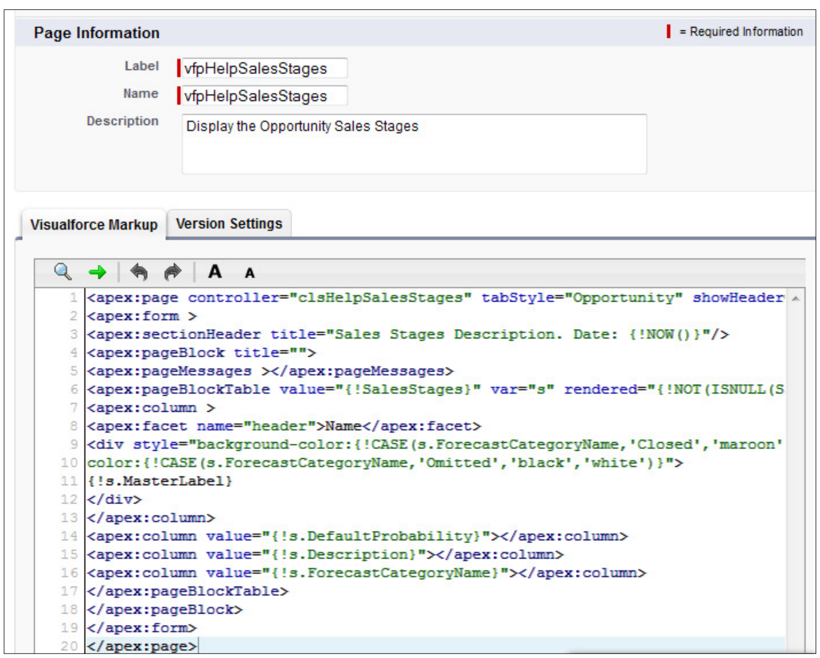


Carry out the following steps to create a Visualforce page to show **Opportunity Sales Stages** descriptions on the home page:

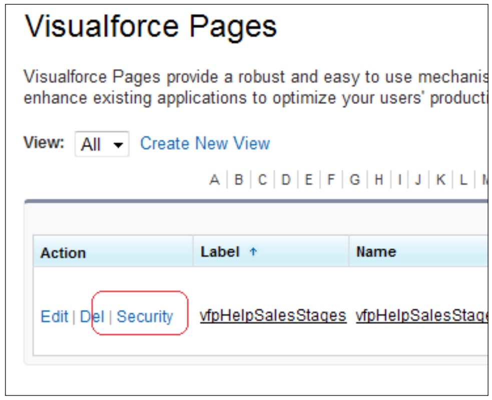
1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `vfpHelpSalesStages` in the **Label** field.
4. Accept the default **vfpHelpSalesStages** in the **Name** field.

5. Paste the following code (as shown in the following screenshot):

```
<apex:page controller="clsHelpSalesStages" tabStyle="Opportunity"
showHeader="false" sidebar="false">
<apex:form >
<apex:sectionHeader title="Sales Stages Description. Date:
{!NOW()}"/>
<apex:pageBlock title="">
<apex:pageMessages ></apex:pageMessages>
<apex:pageBlockTable value="{!SalesStages}" var="s" rendered="{!NO
T(ISNULL(SalesStages))}">
<apex:column >
<apex:facet name="header">Name</apex:facet>
<div style="background-color:{!CASE(s.ForecastCategoryName,'Closed
','maroon','Omitted','#FFA07A','Commit','green','Funnel','blue','g
ray')};
color:{!CASE(s.ForecastCategoryName,'Omitted','black','white')}">
{!s.MasterLabel}
</div>
</apex:column>
<apex:column value="{!s.DefaultProbability}"></apex:column>
<apex:column value="{!s.Description}"></apex:column>
<apex:column value="{!s.ForecastCategoryName}"></apex:column>
</apex:pageBlockTable>
</apex:pageBlock>
</apex:form>
</apex:page>
```



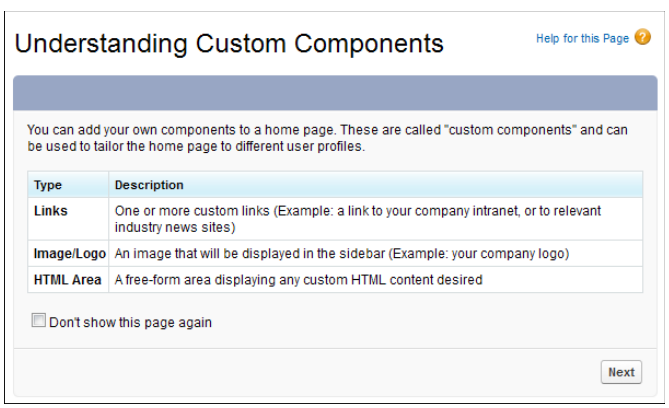
6. Now set security for the required profiles in your organization (this is required for every user profile that you wish to view the sales stage descriptions).



7. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
8. Click on **New**.

[The **New** button is found by scrolling down the page to the **Custom Components** section.]

9. Click on **Next** (on the **Understanding Custom Components** splash screen if shown). The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the **Don't show this page again** checkbox has not previously been checked), as in the following screenshot:



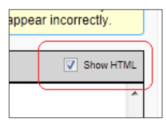
Here, we are presented with the **Step 1. New Custom Components** page.

10. Enter the name of the custom component in the **Name** field. Enter the text `Opportunity Sales Stages`.
11. Select the **HTML Area** option from the **Type** options list.
12. Click on **Next**.
13. Ensure that the option **Wide (Right) Column** is selected within the **Component Position** option list.

> You are unable to change this setting after the component is created.

14. Check the **Show HTML** checkbox.

> The above step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:



15. Paste the following code:

```
<iframe style="width: 100%; height: 400px;" src="/apex/
vfpHelpSalesStages" frameBorder="0"></iframe>
```

16. Click on **Save**.

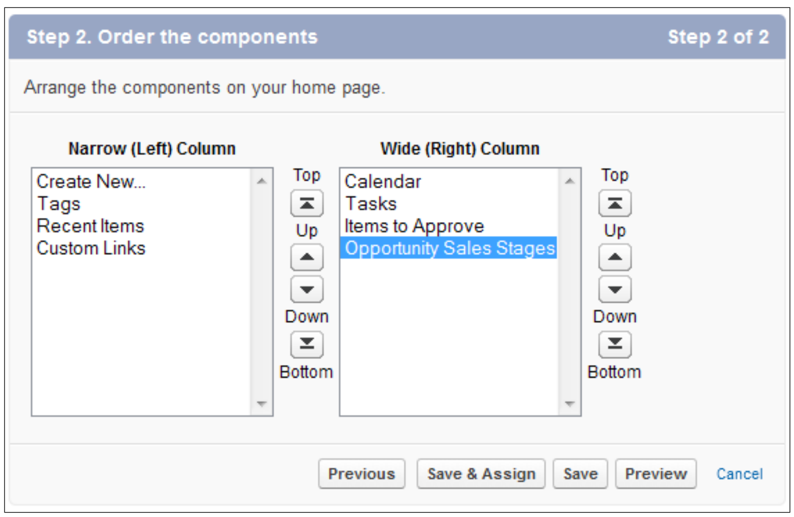
> We have created our **Opportunity Sales Stages** custom home page component but we are not finished yet. We now need to add the custom home page component to a home page layout.

17. Navigate to the home page components going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
18. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**.

    We are presented with the **Step 1. Select the components to show** page.

19. Check the **Opportunity Sales Stages** checkbox in the **Select Wide Components to Show** section, as shown in the following screenshot:

20. Click on **Next**.
21. Move the **Opportunity Sales Stages** to the top position in the **Wide (Right) Column** using the **Arrange the component on your home page.** section, as shown in the following screenshot:



22. Click on **Save**.

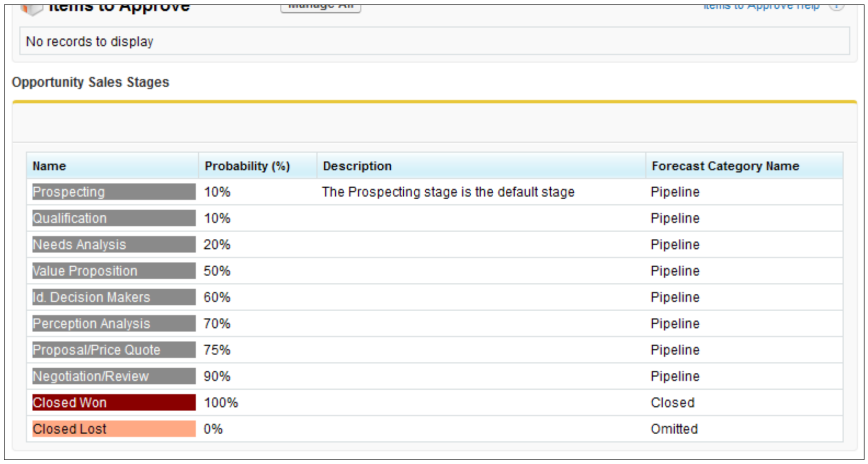
How it works...

By adding the description for the sales stages, the Visualforce and Apex reads the data directly from within Salesforce and so, regardless of whether the sales stage descriptions change or even the sales stages themselves change in the future, the content is always up-to-date.

Users can then see the description, that you would have set, for each of the opportunity stages in the organization on the Visualforce page without having to create dummy opportunities and look at the available stage names.

Salesforce does not natively support the embedding of Visualforce pages into the home page, hence the need to create this HTML custom component. Within this component we are using the `<iframe>` HTML tag we are able to insert the rendered output of a specified Visualforce page.

You can see how this appears on the home page in the following screenshot:


No records to display

Opportunity Sales Stages

Name	Probability (%)	Description	Forecast Category Name
Prospecting	10%	The Prospecting stage is the default stage	Pipeline
Qualification	10%		Pipeline
Needs Analysis	20%		Pipeline
Value Proposition	50%		Pipeline
Id. Decision Makers	60%		Pipeline
Perception Analysis	70%		Pipeline
Proposal/Price Quote	75%		Pipeline
Negotiation/Review	90%		Pipeline
Closed Won	100%		Closed
Closed Lost	0%		Omitted

There's more...

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the component's wizard) you must ensure that the code is valid.

> Pay particular attention to the displayed warning message:
>
> **Please ensure that the HTML code entered below is valid, well formed HTML. Poorly written HTML in this component may cause the entire Home tab to appear incorrectly**

There are a maximum of 20 custom components that can be added to a home page layout.

2

Advanced User Interface

In this chapter, we will cover the following recipes:

- Displaying Case Priority flags using a formula field and salesforce.com images
- Building an Account Revenue indicator using a formula field and custom images
- Creating a clutter-free Account tab using Visualforce
- Showing a tabbed Account Detail page using Visualforce
- Rendering an Account credit score graphically using JavaScript, CSS, and Visualforce
- Presenting an Account credit score graphically using a Google image chart

Introduction

There is a saying that a picture paints a thousand words. Whether this is true, there is no doubt that pictures and images, often, far better describe information than words and numbers alone.

Text and numerical data can often be better represented as graphical charts and images often provide a quick way to compare data values. Images can also provide a more visually powerful style of message delivery for what can otherwise be overlooked static text or numbers.

In this chapter, the recipes are designed to enhance the user interface and provide features and functionality for visually displaying information that is not provided natively within Salesforce.

Displaying Case Priority flags using a formula field and salesforce.com images

The use of certain colors to represent certain states has become commonplace throughout the world. Red generally conveys a warning and on the ubiquitous traffic signal means stop.

Continuing with the look at traffic signals, and the use of red for stop, green for go, and yellow for proceed with caution, these colors are ranked in the order—red, yellow, and green—and using flags of these colors allows us to rank statuses and to highlight certain priorities.

> Salesforce CRM contains accessible images of red, yellow, and green flags.

In this recipe, we will display an image according to a certain priority setting on the Case record detail page where the priority setting is selected using the standard picklist field called **Priority**.

How to do it...

Carry out the following steps to create a formula field to display salesforce.com images:

1. Navigate to the Case customization setup page by going to **Your Name** | **Setup** | **Customize** | **Cases** | **Fields**.

   > Scroll down to the **Case Custom Fields & Relationships** section.

2. Click on **New**.

   > We will be presented with the **Step 1. Choose the field type** page.

3. Select the **Formula** option.
4. Click on **Next**.

   > We will be presented with the **Step 2. Choose output type** page.

5. Enter `Priority Graphic` in the **Field Label** textbox.
6. Click on **Field Name**. When clicking out of the **Field Label** textbox **Field Name** is automatically filled with the value `Priority_Graphic`.
7. Set **Formula Return Type** as **Text**.
8. Click on **Next**.

> We will be presented with the **Step 3. Enter formula** page.

9. Paste the following code:

```
/************************************************************
Priority Graphic for the Case, for High, Medium or Low.
For high priority cases a red flag is displayed; for medium an
orange is shown and for a low a green flag is shown.
************************************************************/
IMAGE
(
 IF( ISPICKVAL(Priority,"Low"),
   "/img/samples/flag_green.gif",
     IF( ISPICKVAL(Priority,"Medium"),
       "/img/samples/flag_yellow.gif",
         "/img/samples/flag_red.gif"
     )
 ),
"", 15, 15
)
```

10. In the **Blank Field Handling** section, select the **Treat blank fields as blanks** option.
11. Click on **Next**.

> We will be presented with the **Step 4. Establish field-level security** page.

12. Select the profiles to which you want to grant read access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
13. Click on **Next**.

> We will be presented with the **Step 5. Add to page layouts** page.

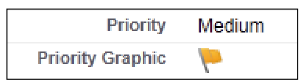
14. Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
15. Finally, click on **Save**.

How it works...

The formula field graphic is dynamically generated based on the selected priority value and the rendered image appears on the case detail page.

You can see what this looks like when the **Priority** is set to **High** in the following screenshot:

You can see what this looks like when the **Priority** is set to **Medium** in the following screenshot:



You can see what this looks like when the **Priority** is set to **Low** in the following screenshot:

Building an Account Revenue indicator using a formula field and custom images

Increasingly on the Web, we find websites that use images of gold or silver stars to provide reviews and to rank the quality or usefulness of various products and services.

It has become universally accepted that one or no star equates to something very poor and a rating of five stars is seen to be excellent. By building an incremental number of images, we can create an associated image list of, say 1 to 5, that conveys a rating and ranking factor.

For this recipe, we are using a dollar image that will be repeated depending on the value of the **Account Revenue** amount.

The dollar image we are using is a custom image and is not provided by the Salesforce CRM application.

Using the value entered in the standard **Account Revenue** field we will create a custom formula field to build a set of images, from one to five, whenever the **Account Revenue** amount meets a certain threshold criteria.

The thresholds that will formulate are:

- Greater than (or equal to) $100,000 = one dollar image
- Greater than (or equal to) $500,000 = two dollar images
- Greater than (or equal to) $1 million = three dollar images
- Greater than (or equal to) $2 million = four dollar images
- Greater than (or equal to) $5 million = five dollar images

Amounts less than $100,000 will have no images displayed.

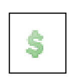
Getting ready

We can use custom images that we have either created ourselves or obtained from an external source and store them in the Salesforce CRM application.

External images can be uploaded into Salesforce CRM by carrying out the following steps:

1. Create or source a suitable image to represent a dollar symbol.

The image we have used here is a 16-by-16 pixel icon called `money_dollar.png` available from famfamfam.com, shown as follows:



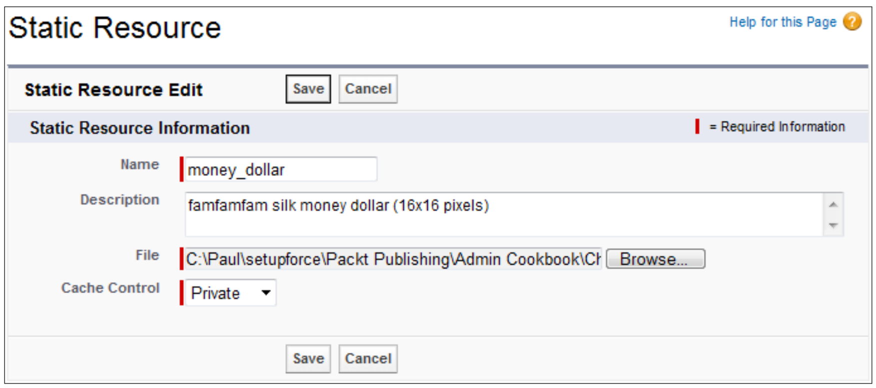
FamFamFam have provided various images from their "Silk Icons library" available under the Creative Commons Attribution 2.5 License at `http://www.famfamfam.com/lab/icons/silk/`.

2. Navigate to the Static Resources setup page by going to **Your Name** | **Setup** | **Develop** | **Static Resources**.
3. Click on **New**.

4. Enter the name of the Static Resource in the **Name** field. For this recipe type the text `money_dollar`.
5. In the File uploader control, click on the **Browse...** button to select the image to upload from your computer. In this recipe choose the image identified in step 1.

> [Ignore the Cache Control picklist selection and leave it as default **Private** (Cache Control is only relevant to static resources used in Force.com sites).]

The Static Resources setup screen appears, shown as follows:

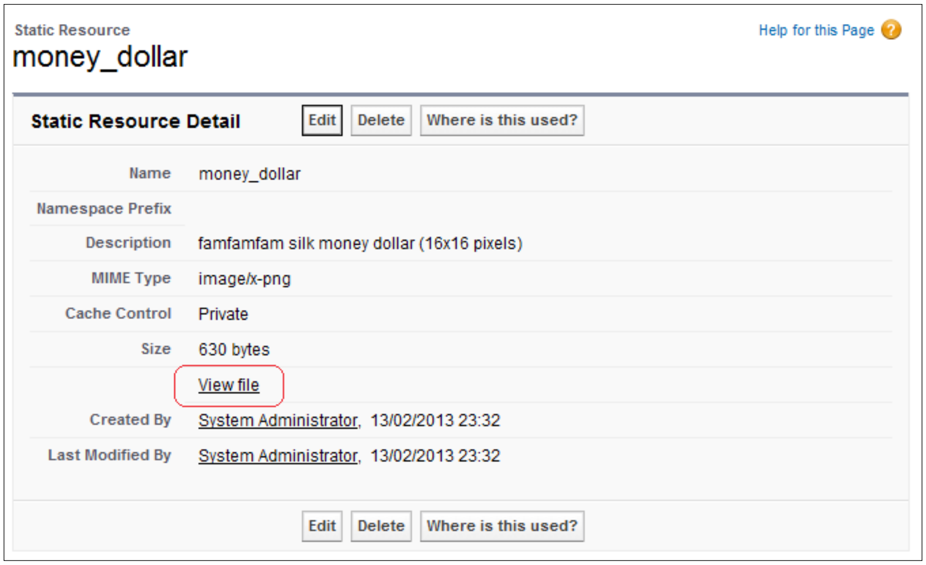


6. Click on **Save**.

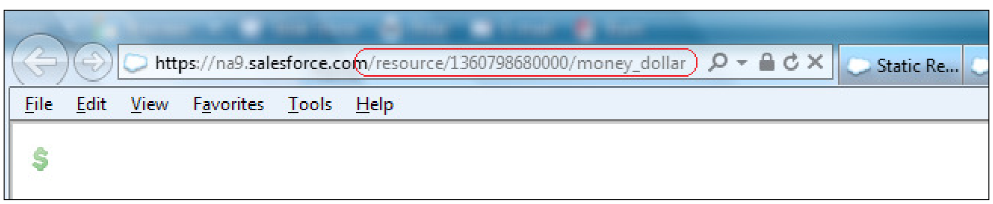
> [Static resources allow us to upload images that we can reference within Salesforce CRM, such as from formula fields, Visualforce pages, and so on. This reference is a Web URL and is formed as `/resource/[UNIQUE ID]/money_dollar` where the `[UNIQUE ID]` is a unique ID which is generated for every static resource and is unique throughout every Salesforce CRM system.]

Now, to find the ID for the static resource uploaded above, carry out the following steps:

1. Click on **View File**, as shown in the following screenshot:



2. Note the Web URL that is displayed in the browser address bar; this is the ID for the static resource, as shown in the following screenshot:



> The URL that is generated is `/resource/1360798680000/money_dollar`.
>
> You will now need to make a note of the URL that is shown in your Salesforce organization. This URL is specified for the reference to the image in a custom formula field that is used in this recipe using the steps in the following section.

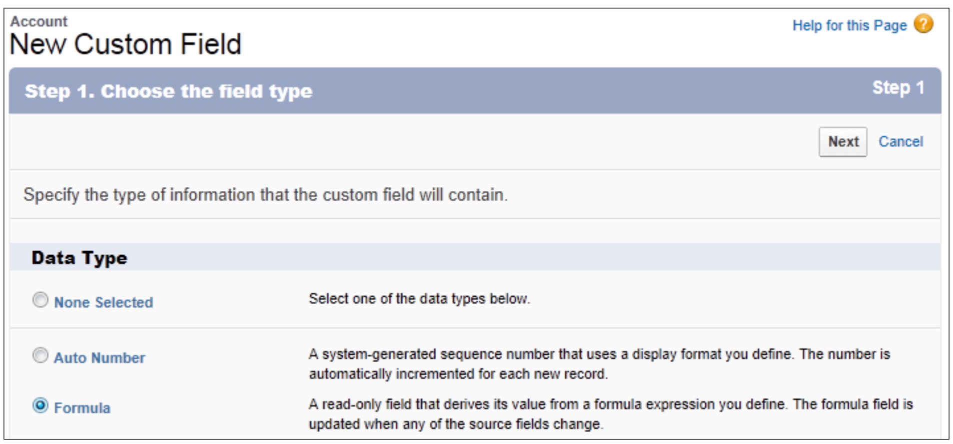
How to do it...

Now let's create the custom field that will reference the dollar image that was uploaded above:

1. Navigate to the Account customization setup page by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Fields**.
2. Click on **New**.

> We will be presented with the **Step 1. Choose the field type** page.

3. Select the **Formula** option, as shown in the following screenshot:



4. Click on **Next**.

> We will be presented with the **Step 2. Choose output type** page.

5. Enter `Account Revenue Graphic` in the **Field Label** textbox.
6. Click on **Field Name**. When clicking out of the **Field Label** textbox **Field Name** is automatically filled with the value `Account_Revenue_Graphic`.
7. Set the **Formula Return Type** as **Text**.
8. Click on **Next**.

> We will be presented with the **Step 3. Enter formula** page.

9. Paste the following code in the formula edit box (as shown in the image further down):

> Remember to replace the URL shown with the URL from your Salesforce organization.

```
/*********************************************************
Begin the check for Annual Revenue value and set the following:
Greater than (or equal to) 100,000 = One Dollar image
Greater than (or equal to) 500,000 = Two Dollar image
Greater than (or equal to) 1,000,000 = Three Dollar image
Greater than (or equal to) 2,000,000 = Four Dollar image
Greater than (or equal to) 5,000,000 = Five Dollar image
*********************************************************/
IF( AnnualRevenue > 99999, IMAGE("/resource/1360798680000/money_
dollar", "$", 16, 16),"")
&
IF( AnnualRevenue > 499999, IMAGE("/resource/1360798680000/money_
dollar", "$", 16, 16),"")
&
IF( AnnualRevenue > 999999, IMAGE("/resource/1360798680000/money_
dollar", "$", 16, 16),"")
&
IF( AnnualRevenue > 1999999, IMAGE("/resource/1360798680000/money_
dollar", "$", 16, 16),"")
&
IF( AnnualRevenue > 4999999, IMAGE("/resource/1360798680000/money_
dollar", "$", 16, 16),"")
```

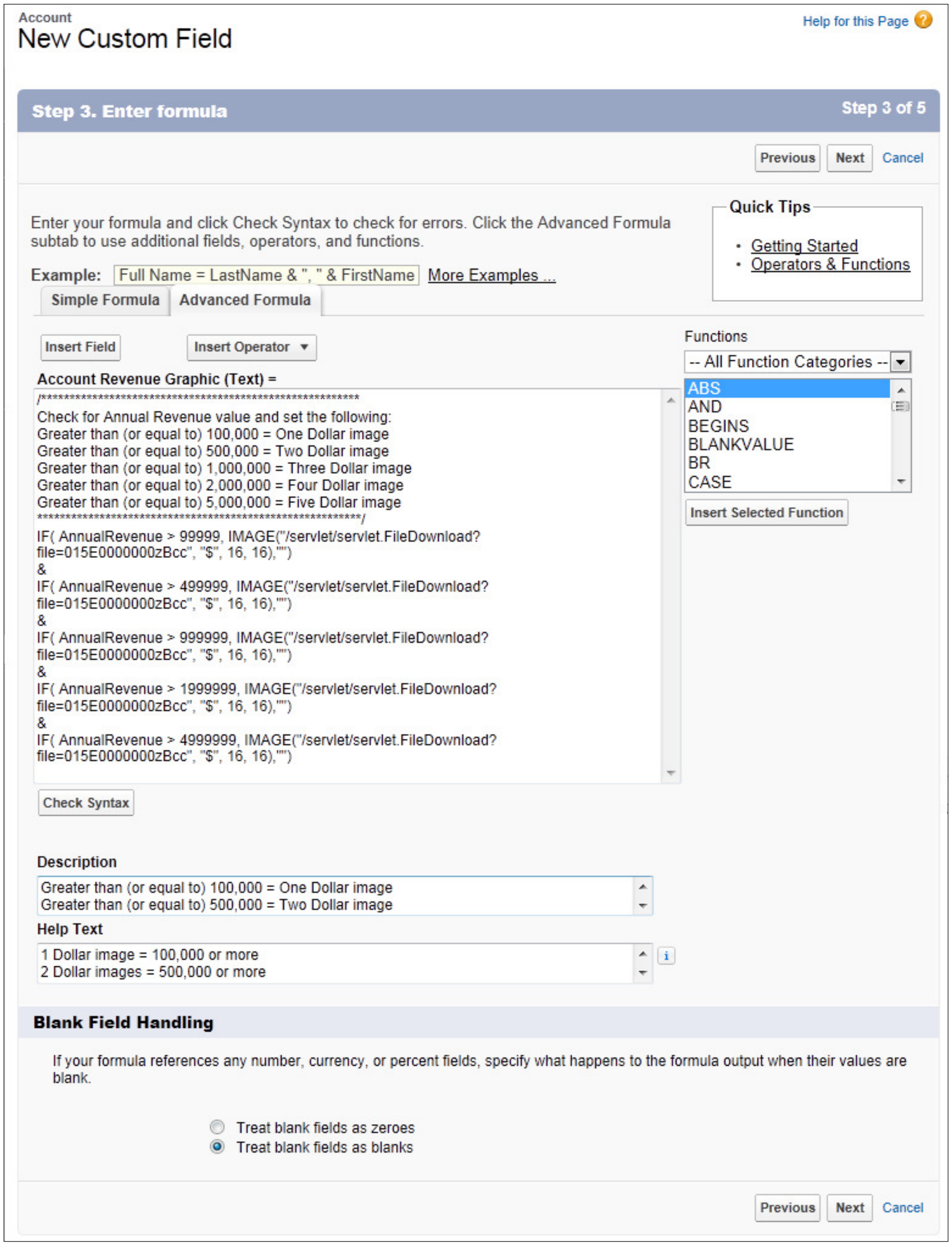
10. Optionally enter the following in the **Description** field:

```
Greater than (or equal to) 100,000 = One Dollar image
Greater than (or equal to) 500,000 = Two Dollar image
Greater than (or equal to) 1,000,000 = Three Dollar image
Greater than (or equal to) 2,000,000 = Four Dollar image
Greater than (or equal to) 5,000,000 = Five Dollar image
```

11. Optionally enter the following in the **Help Text** field:

```
1 Dollar image = 100,000 or more
2 Dollar images = 500,000 or more
3 Dollar images = 1,000,000 or more
4 Dollar images = 2,000,000 or more
5 Dollar images = 5,000,000 or more
```

12. In the **Blank Field Handling** section, select the **Treat blank fields as blanks** option, as shown in the following screenshot:



13. Click on **Next**.

> We will be presented with the **Step 4. Establish field-level security** page.

14. Select the profiles to which you want to grant read access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
15. Click on **Next**.

> We will be presented with the **Step 5. Add to page layouts** page.

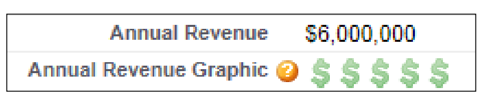
16. Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
17. Click on **Save**.

How it works...

The formula field graphic is dynamically generated based on the annual revenue value and the rendered image appears on the **Account Detail** page.

You can see what this looks like when the **Annual Revenue** is set to **$1,000,000** in the following screenshot:

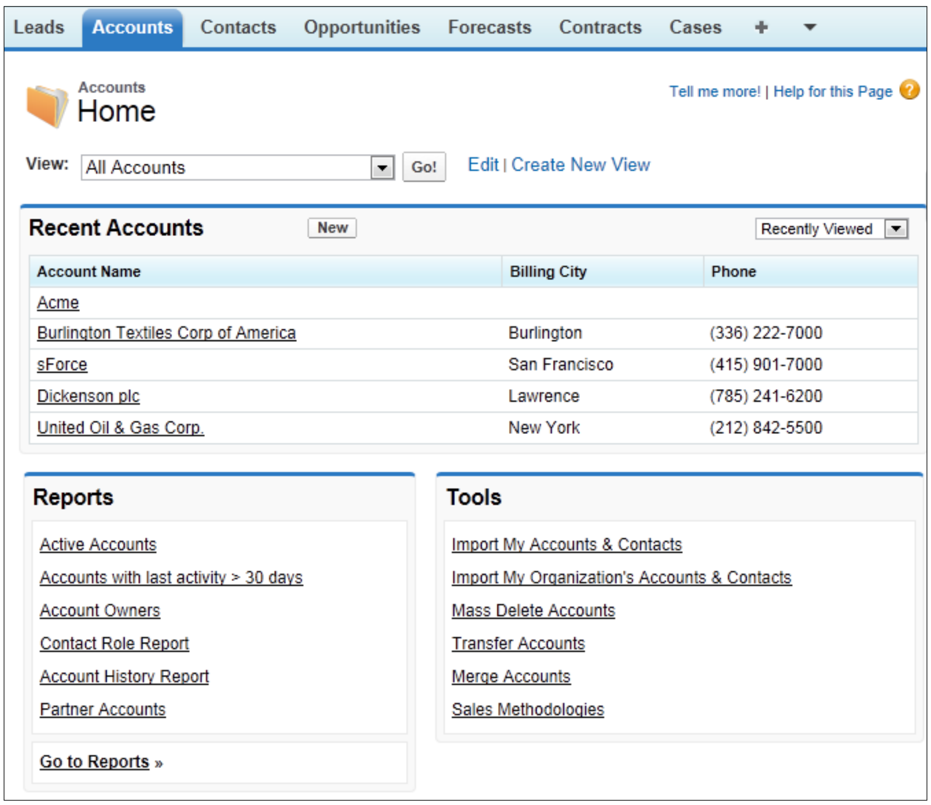
You can see what this looks like when the **Annual Revenue** is set to **$6,000,000** in the following screenshot:



Creating a clutter-free Account tab using Visualforce

When users click on the **Account** tab in Salesforce CRM they are presented with a screen that shows by default a **Recent Accounts** section within which is a list of their recently viewed accounts.

Below the **Recent Accounts** section are **Reports** and **Tools** sections, as shown in the following screenshot:



This screen can be very useful when users are aware of its capabilities, however less-experienced users are often confused when they click on the **Account** tab and are presented with a list of recently viewed accounts.

Users are sometimes concerned that they have "lost" their accounts as they expect to see them all when clicking on the **Accounts** tab and can be a little frustrated to have to click on the button marked **Go!** every time they want a list of all their accounts.

In this recipe you will see how you can override the standard **Account** tab's page and directly show a simple complete list of accounts owned by users (as part of an appropriate list view).

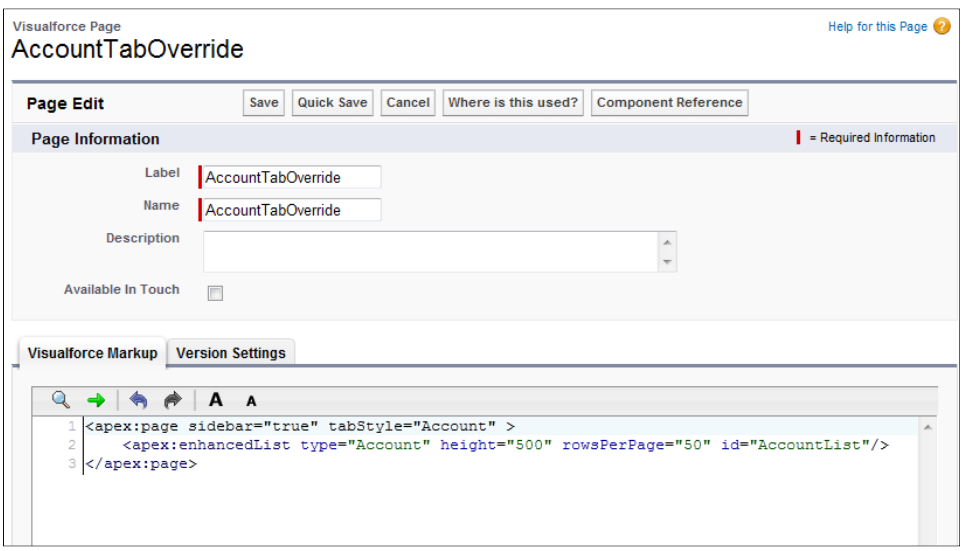
Not only does this recipe save users' time but it also de-clutters the screen by removing all the reports and tools links from the Salesforce Account tab's page.

How to do it...

Carry out the following steps to create a Visualforce page:

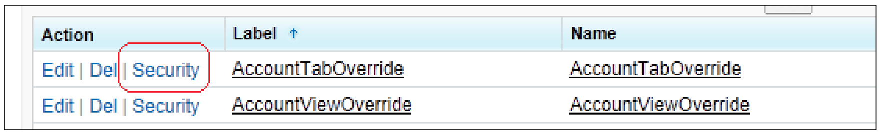
1. Navigate to the Visualforce Pages setup screen by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `AccountTabOverride` in the **Label** field.
4. Accept the default **AccountTabOverride** in the **Name** field.
5. Paste the following code (as shown in the screenshot further down):

```
<apex:page sidebar="true" tabStyle="Account" >
    <apex:enhancedList type="Account" height="500"
rowsPerPage="50" id="AccountList"/>
</apex:page>
```

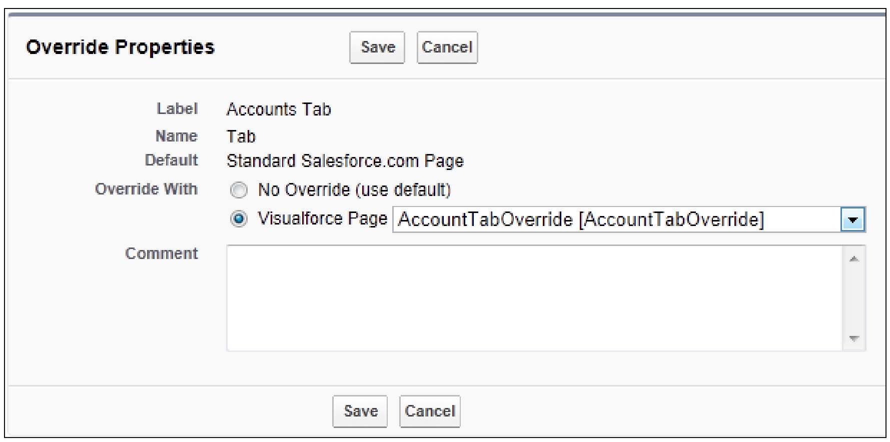


6. Click on **Save**.
7. Now set the Security for the required profiles in your organization by navigating to **Your Name** | **Setup** | **Develop** | **Pages**.

8. Locate the row for the Visualforce Page **AccountTabOverride** and click the **Security** link, as shown in the following screenshot:



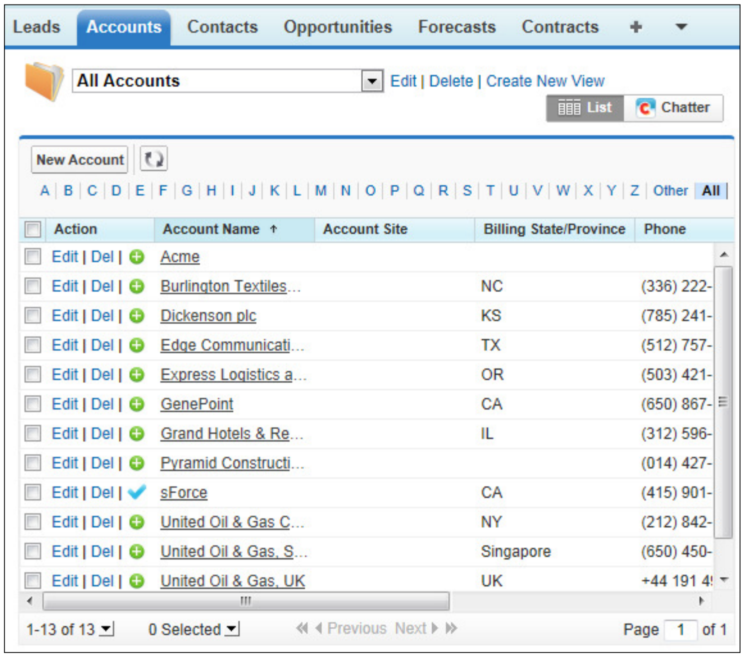
9. Set the security for the required profiles.
10. Now, override the standard **Account** tab by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Buttons and Links**.
11. On the **Accounts Buttons and Links** page locate the **Accounts Tab** row and click on **Edit**.
12. Click on the **Override With** option and select **Visualforce Page**.
13. Select the **AccountTabOverride** Visualforce page as shown in the following screenshot:



14. Click on **Save**.

How it works...

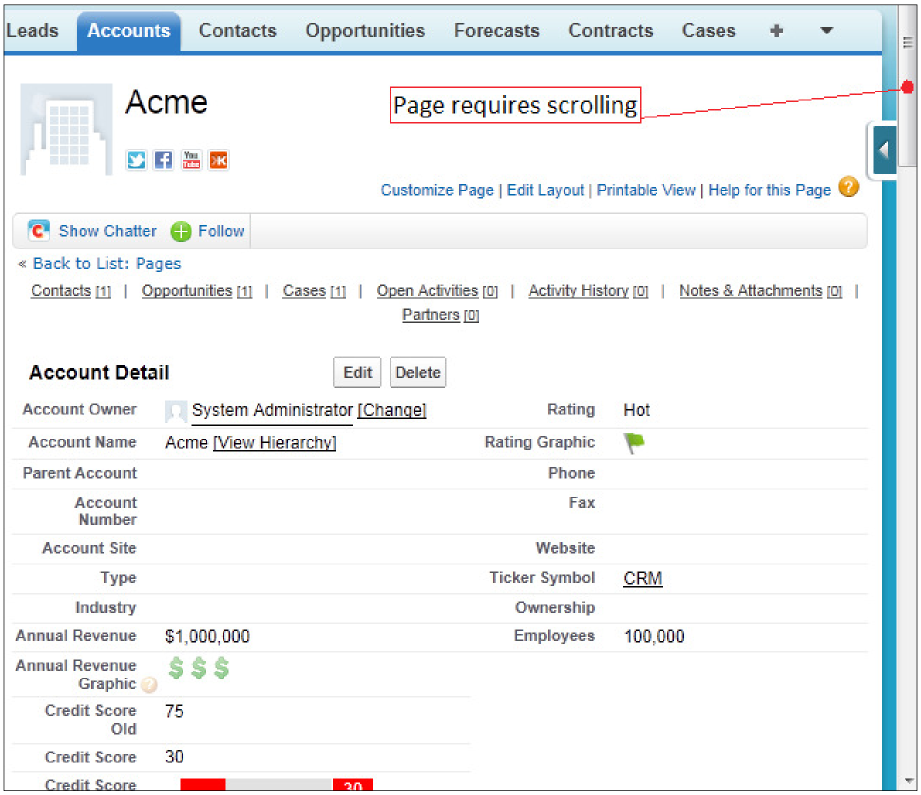
When users click on the **Account** tab, there is a clutter-free page as shown in the following screenshot:



Showing a tabbed Account Detail page using Visualforce

Salesforce CRM provides related lists on the standard pages which are added vertically to the lower section of the page below the detail section.

As more and more related lists are added for an object, the sections of related list data can make the page stretch vertically and make it a challenge for users to scroll down to the page to locate the required information as shown in the following screenshot:



To counter this, Salesforce provides hover links at the top of the page that allow the users to navigate via hyperlinks directly to the required related list selection further down the page. This is very useful when users are aware and dexterous enough to use its capabilities; however, some users experience navigating pages that have many related lists.

In this recipe, we will look at how you can show a Visualforce page to override the standard **Account** view and present the related list as separate tabs instead of having them rendered at the bottom of the page.

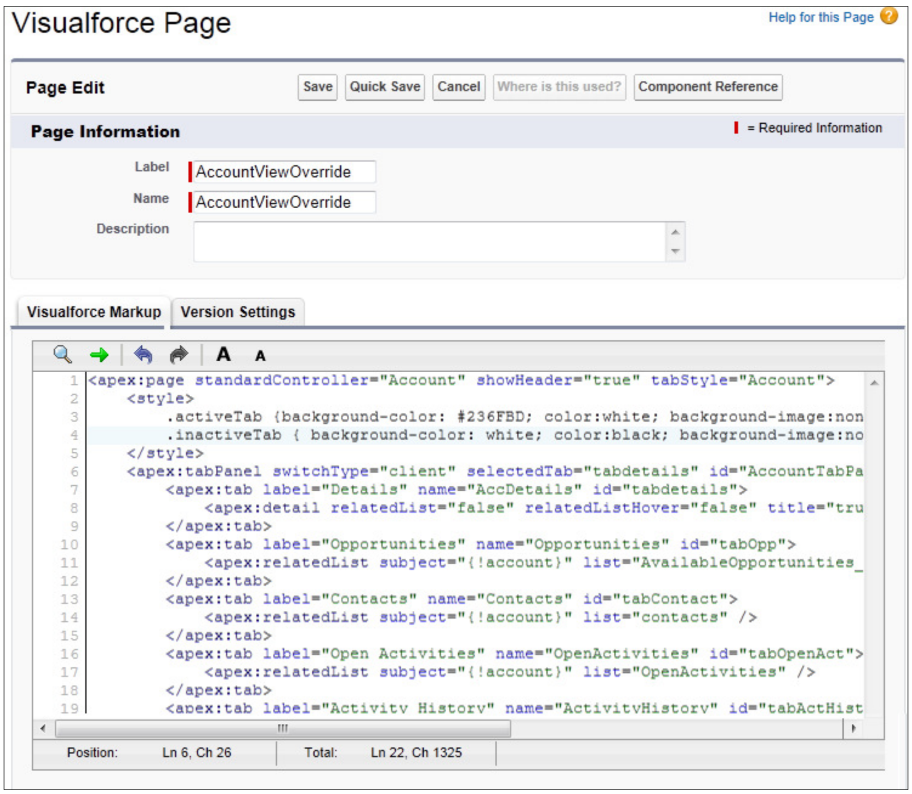
How to do it...

Carry out the following steps to create a Visualforce page:

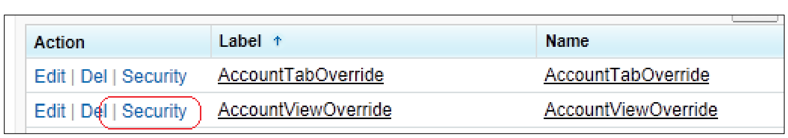
1. Navigate to the Visualforce Pages setup screen by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `AccountViewOverride` in the **Label** field.
4. Accept the default **AccountViewOverride** in the **Name** field.
5. Paste the following code in the formula edit box (as shown in the screenshot further below):

```
<apex:page standardController="Account" showHeader="true"
tabStyle="Account">
  <style>
    .activeTab {background-color: #236FBD; color:white;
background-image:none}
    .inactiveTab { background-color: white; color:black;
background-image:none}
  </style>

   <apex:tabPanel switchType="client" selectedTab="tabdetails"
id="AccountTabPanel" tabClass="activeTab" inactiveTabClass="inacti
veTab">
      <apex:tab label="Details" name="AccDetails" id="tabdetails">
    <apex:detail relatedList="false" relatedListHover="false"
title="true" inlineEdit="false"/>
      </apex:tab>
    <apex:tab label="Available Opportunities" name="Available
Opportunities" id="tabAvOpp">
        <apex:relatedList subject="{!account}" list="AvailableOppo
rtunities__r" />
    </apex:tab>
    <apex:tab label="Contacts" name="Contacts" id="tabContact">
        <apex:relatedList subject="{!account}" list="contacts" />
    </apex:tab>
      <apex:tab label="Open Activities" name="OpenActivities"
id="tabOpenAct">
         <apex:relatedList subject="{!account}"
list="OpenActivities" />
      </apex:tab>
      <apex:tab label="Activity History" name="ActivityHistory"
id="tabActHist" >
         <apex:relatedList subject="{!account}"
list="ActivityHistories" pageSize="15" />
```

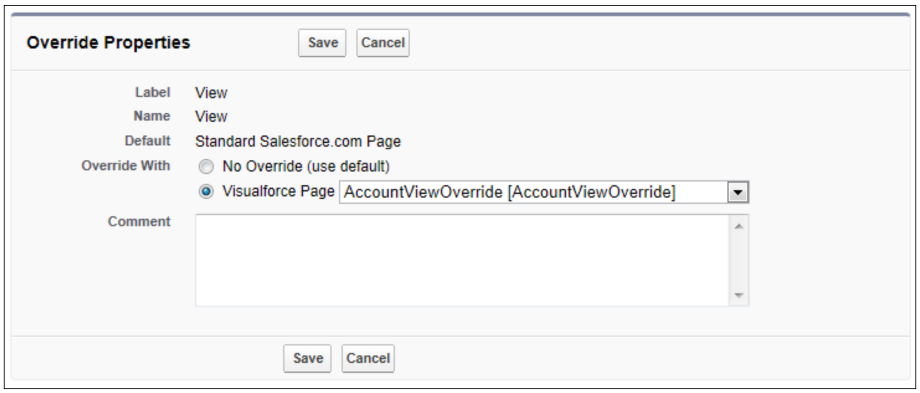
```
        </apex:tab>
        <apex:tab label="Notes and Attachments"
name="NotesAndAttachments" id="tabNews">
            <apex:relatedList subject="{!account}"
list="NotesAndAttachments" />
        </apex:tab>
    </apex:tabPanel>
</apex:page>
```



6. Now set the Security for the required profiles in your organization by going to **Your Name** | **Setup** | **Develop** | **Pages**.
7. Locate the row for the Visualforce Page **AccountViewOverride** and click on the **Security** link as shown in the following screenshot:



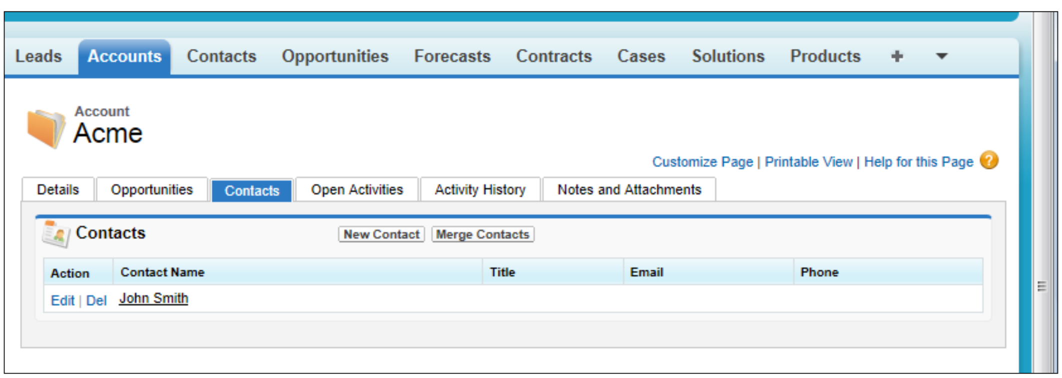
8. Now, override the standard **Account View** by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Buttons and Links**.
9. On the **Accounts Buttons and Links** page locate the **Accounts View** row and click on **Edit**.
10. Click on the **Override With** option and select **Visualforce Page**.
11. Select the **AccountViewOverride** Visualforce page as shown in the following screenshot:



12. Click on **Save**.

How it works...

When users click on an account record, a tabbed page is presented as shown in the following screenshot:



Rendering an Account credit score graphically using JavaScript, CSS, and Visualforce

Here we describe a recipe for rendering a credit score graphically using JavaScript and CSS within a Visualforce page and adding the page as an inline section on a Salesforce Account Page Layout. First, we create a custom Account credit score which will feed the values rendered in the graphical range.

Getting ready

Carry out the following steps to create a custom **Credit Score** field on the **Account** object:

1. Navigate to the Account Fields setup page by going to **Your Name** | **Customize** | **Accounts** | **Fields**.

   > Scroll down to the **Account Custom Fields & Relationships** section.

2. Click on **New**.

   > We will be presented with the **Step 1. Choose the field type** page.

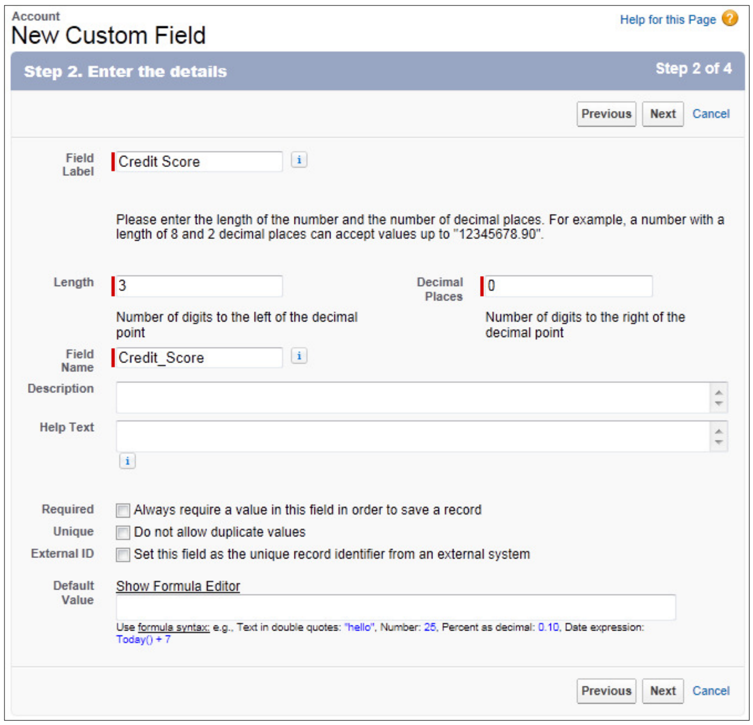
3. Choose **Number** from the **Data Type** options.
4. Click on **Next**.

   > We will be presented with the **Step 2. Enter the details** page.

5. Enter `Credit Score` in **Field Label**.

6. Enter `3` in the **Length** field.

Accept the default option of **0** in the **Decimal Places** field (as shown in the following screenshot).



7. Click on **Next**.

We will be presented with the **Step 3. Establish field-level security** page.

8. Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.

9. Click on **Next**.

> We will be presented with the **Step 4. Add to page layouts** page. Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.

10. Finally click on **Save**.

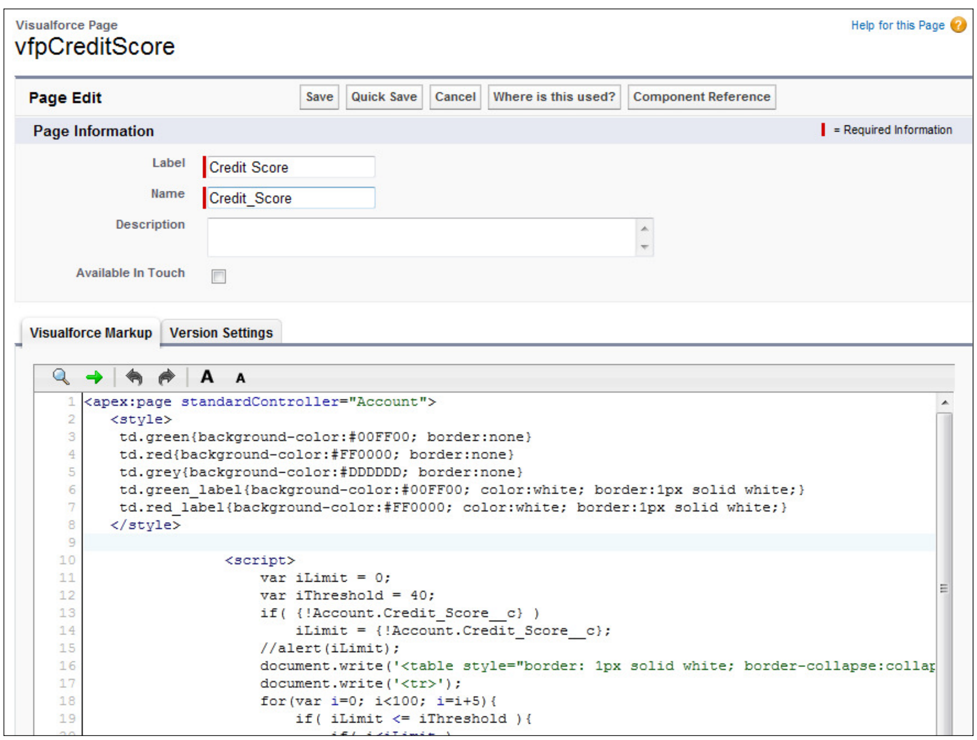
How to do it...

Carry out the following steps to create a Visualforce page:

1. Navigate to the Visualforce Pages setup screen by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `Credit Score` in the **Label** field.
4. Accept the default **Credit_Score** in the **Name** field.
5. Paste the following code (as shown in the screenshot further below):

```
<apex:page standardController="Account">
   <style>
    td.green{background-color:#00FF00; border:none}
    td.red{background-color:#FF0000; border:none}
    td.grey{background-color:#DDDDDD; border:none}
    td.green_label{background-color:#00FF00; color:white;
border:1px solid white;}
    td.red_label{background-color:#FF0000; color:white; border:1px
solid white;}
   </style>
                <script>
                    var iLimit = 0;
                    var iThreshold = 40;
                    if( {!Account.Credit_Score__c} )
                        iLimit = {!Account.Credit_Score__c};
                    document.write('<table style="border: 1px
solid white; border-collapse:collapse;">');
                    document.write('<tr>');
                    for(var i=0; i<100; i=i+5){
                        if( iLimit <= iThreshold ){
                            if( i<iLimit )
```

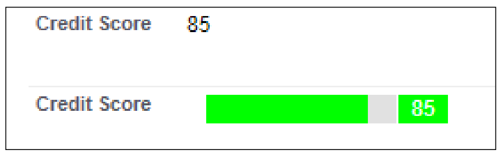
```
                                document.write('<td class="red"
nowrap="nowrap"> </td>');
                            else
                                document.write('<td class="grey"
nowrap="nowrap"> </td>');
                        }else{
                            if( i<iLimit )
                                document.write('<td class="green"
nowrap="nowrap"> </td>');
                            else
                                document.write('<td class="grey"
nowrap="nowrap"> </td>');
                        }
                    }
                    iLimit <= iThreshold ? document.write('<td
class="red_label" nowrap="nowrap">  ' + iLimit +
'  </td></tr></table>') :
                        document.write('<td class="green_label"
nowrap="nowrap">  ' + iLimit + '  </td></tr></
table>');
                </script>
</apex:page>
```



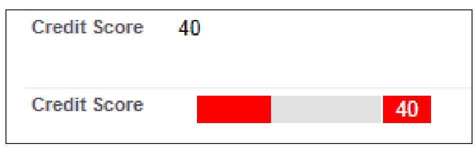
6. Now set the Security for the required profiles in your organization by going to **Your Name** | **Setup** | **Develop** | **Pages**.
7. Locate the row for the Visualforce Page **Credit Score** and click on the **Security** link.

How it works...

Navigate to an **Account Detail** page and enter the value `85` in the custom **Credit Score** number field. Upon saving this, the **Account Detail** page displays the **Credit Score** image as shown in the following screenshot:



Navigate to the **Account Detail** page (as mentioned earlier) and enter the value `40` in the **Credit Score** number field. Upon saving this, the **Account Detail** page displays the **Credit Score** image as shown in the following screenshot:



Presenting an Account credit score graphically using a Google image chart

This recipe presents the steps for rendering a credit score graphically using a Google Chart contained within a custom Salesforce CRM formula field. Here we are using a dial-type chart from Google called a **Google-o-meter chart**. The Google-o-meter is a gauge that points toward a single value on a range. More details can be found at `https://developers.google.com/chart/image/docs/gallery/googleometer_chart#introduction`.

First, we create a custom account credit score which will feed the values rendered in the graphical range.

Getting ready

To create a custom **Credit Score** field on the account, follow these steps:

1. Navigate to the Account Fields setup page by going to **Your Name** | **Customize** | **Accounts** | **Fields**.

[Scroll down to the Account **Custom Fields & Relationships** section.]

2. Click on **New**.

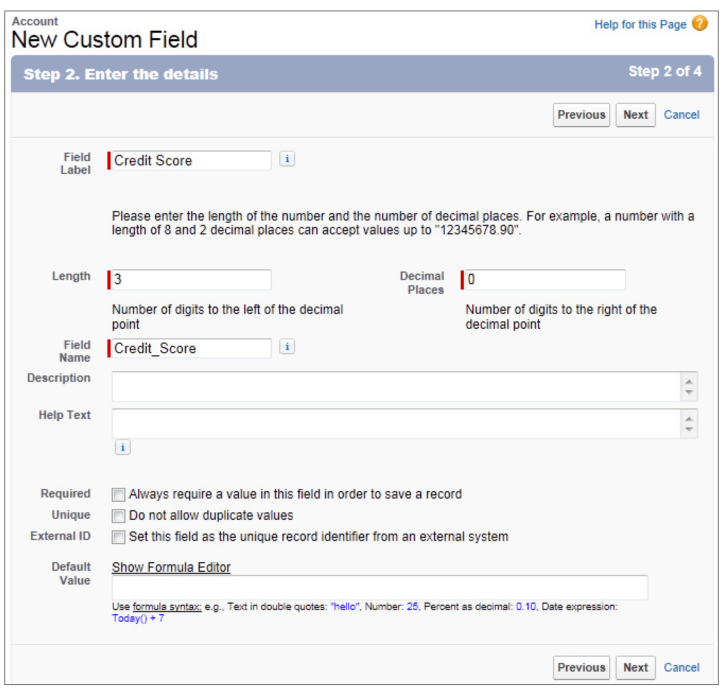
[We will be presented with the **Step 1. Choose the field type** page.]

3. Choose **Number** from the **Data Type** options.
4. Click on **Next**.

[We will be presented with the **Step 2. Enter the details** page.]

5. Enter `Credit Score` in **Field Label**.
6. Enter `3` in the **Length** field.

[Accept the default option of `0` in the **Decimal Places** field (as shown in the followings screenshot).]



7. Click on **Next**.

> We will be presented with the **Step 3. Establish field-level security** page.

8. Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
9. Click on **Next**.

> We will be presented with the **Step 4. Add to page layouts** page.

10. Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
11. Finally click on **Save**.

How to do it...

Carry out the following steps to create a formula field to display a Google chart:

1. Navigate to the Account Fields setup page by going to **Your Name** | **Customize** | **Accounts** | **Fields**.

> Scroll down to the **Account Custom Fields & Relationships** section.

2. Click on **New**.

> We will be presented with the **Step 1. Choose the field type** page.

3. Select the **Formula** option.

4. Click on **Next**.

> We will be presented with the **Step 2. Choose output type** page.

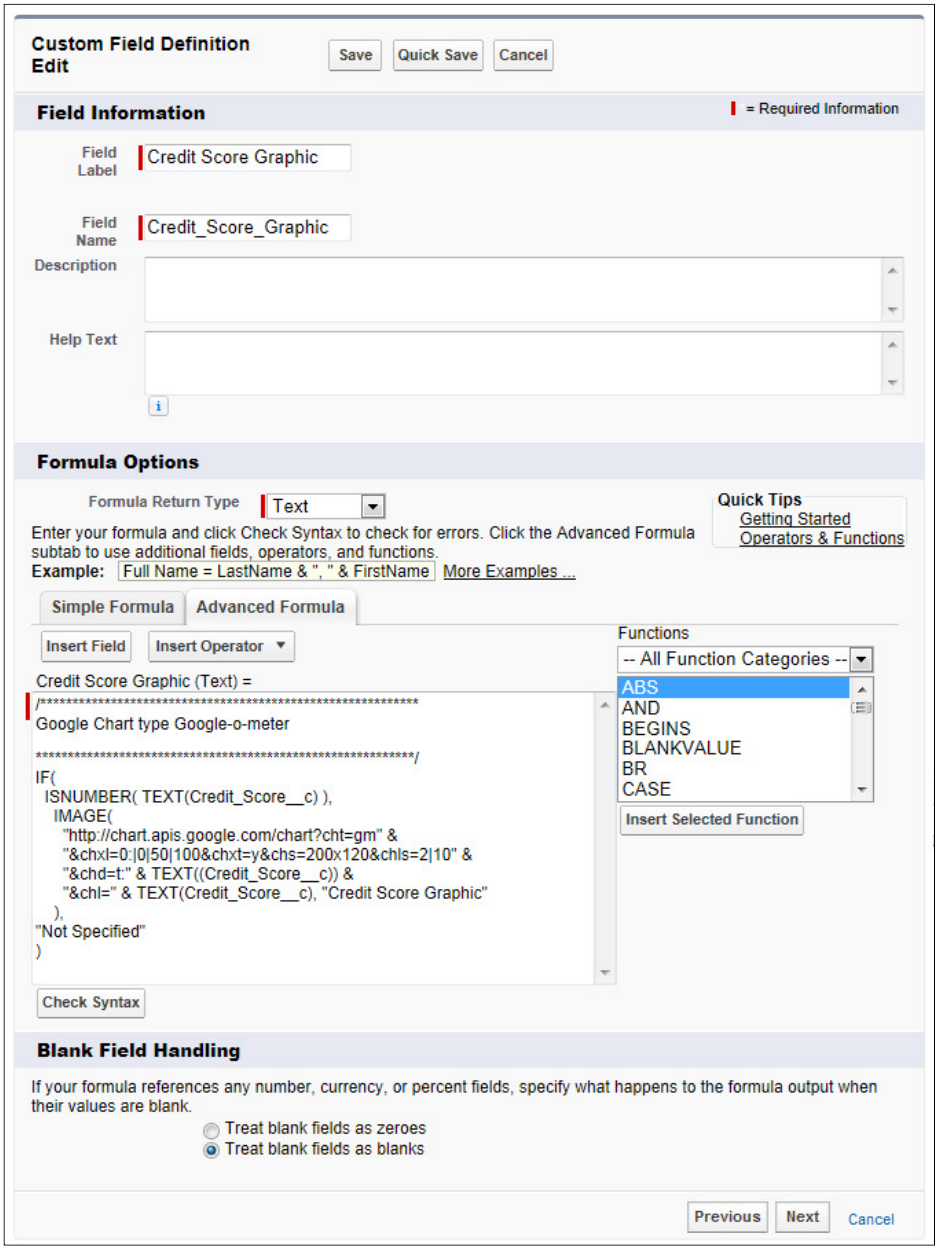
5. Enter `Credit Score Graphic` in the **Field Label** textbox.
6. Click on **Field Name**. When clicking out of the **Field Label** textbox **Field Name** is automatically filled with the value `Graphic_Score_Graphic`.
7. Set the **Formula Return Type** as **Text** (as shown in the following screenshot)
8. Click on **Next**.

> We will be presented with the **Step 3. Enter formula** page.

9. Paste the following code in the formula edit box (as shown in the screenshot further below):

```
/***********************************************************
Google Chart type Google-O-meter

***********************************************************/
IF(
  ISNUMBER( TEXT(Credit_Score__c) ),
    IMAGE(
      "http://chart.apis.google.com/chart?cht=gm" &
      "&chxl=0:|0|50|100&chxt=y&chs=200x120&chls=2|10" &
      "&chd=t:" & TEXT((Credit_Score__c)) &
      "&chl=" & TEXT(Credit_Score__c), "Credit Score Graphic"
    ),
"Not Specified"
)
```

10. In the **Blank Field Handling** section, select the option **Treat blank fields as blanks** as shown in the following screenshot:



11. Click on **Next**.

> We will be presented with the **Step 4. Establish field-level security** page.

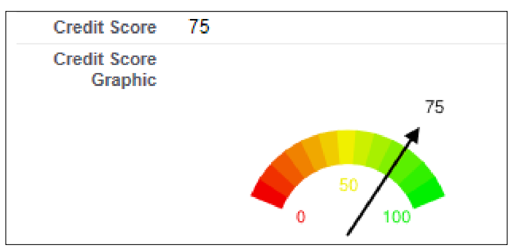
12. Select the profiles to which you want to grant edit access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
13. Click on **Next**.

> We will be presented with the **Step 5. Add to page layouts** page.

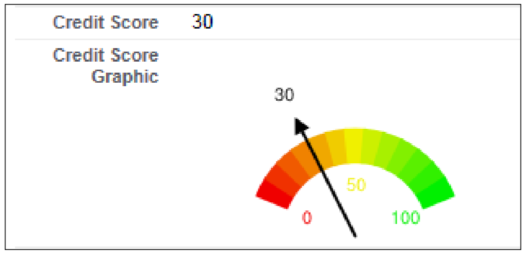
14. Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.
15. Finally, click on **Save**.

How it works...

Navigate to an **Account Detail** page and enter the value `75` in the custom **Credit Score** number field. Upon saving this, the **Account Detail** page displays the **Credit Score Graphic** image as shown in the following screenshot:



Navigate to the **Account Detail** page (as above) and enter the value `30` in the **Credit Score** number field. Upon saving this, the **Account Detail** page displays the **Credit Score Graphic** image as shown in the following screenshot:



Describing each part of the Google Chart code in the formula field gives:

```
/************************************************************
Google Chart type Google-O-meter

************************************************************/
IF(
  ISNUMBER( TEXT(Credit_Score__c) ),
    IMAGE(
      "http://chart.apis.google.com/chart?cht=gm" &
      "&chxl=0:|0|50|100&chxt=y&chs=200x120&chls=2|10" &
      "&chd=t:" & TEXT((Credit_Score__c)) &
      "&chl=" & TEXT(Credit_Score__c), "Credit Score Graphic"
    ),
"Not Specified"
)
```

The following comment section describes the code in the formula:

```
/************************************************************
Google Chart type Google-O-meter

************************************************************/
```

The following code snippet checks that the Credit Score contains a number. If so then it continues to build the Google Chart code. If there is no valid number then it returns the value `"Not Specified"`.

```
IF(
  ISNUMBER( TEXT(Credit_Score__c) ),
<.........................................>
"Not Specified"
)
```

The following code snippet is for Google Chart image construction:

```
IMAGE(
  "http://chart.apis.google.com/chart?cht=gm" &
  "&chxl=0:|0|50|100&chxt=y&chs=200x120&chls=2|10" &
  "&chd=t:" & TEXT((Credit_Score__c)) &
  "&chl=" & TEXT(Credit_Score__c), "Credit Score Graphic"
),
```

Use the Google Chart URL:

```
http://chart.apis.google.com/chart?
```

The following line specifies the chart type `gm` (Google-O-meter):

```
cht=gm
```

The following line sets the labels for the chart:

```
chxl=0:|0|50|100
```

The following line specifies using the Y-axis:

```
chxt=y
```

The following line sets the dimensions for the chart (width x height):

```
chs=200x120
```

The following line specifies the arrow line width and arrow head (2px line and small arrow head):

```
chls=2|10
```

The following line sets the data value passed to the graph (the data from the **Credit Score** field is passed):

```
chd=t:" & TEXT((Credit_Score__c))
```

The following line specifies the data label on the chart (the data from the **Credit Score** field is passed):

```
chl=" & TEXT(Credit_Score__c)
```

There's more...

For more information to create other types of Google charts, navigate to:

`http://imagecharteditor.appspot.com/`

> Accessing Google Charts requires sending data from Salesforce over the Internet, so it is not secure. You should ensure that only non-sensitive data is being sent.

In Internet Explorer, there is security built in to check for HTTP/HTTPS mixed content. Salesforce uses HTTPS which is secure and as described earlier the Google Chart website uses HTTP which is not secured using SSL encryption. Internet Explorer may generate a warning for each user's browser which you can suppress using the following tip.

How do I suppress browser security warnings in Internet Explorer?

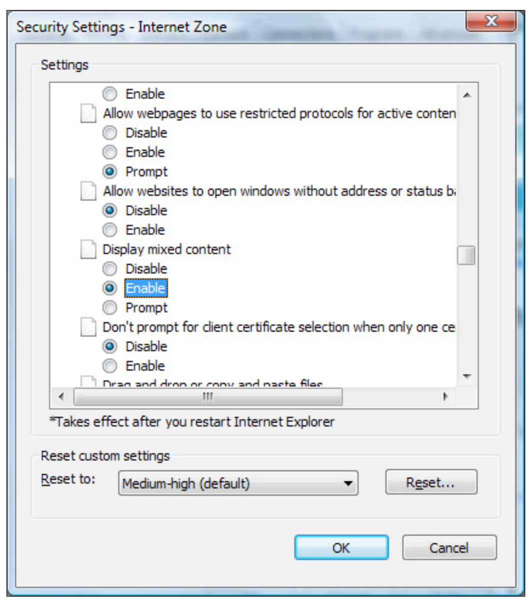
By default, Internet Explorer displays the following security warning message when a page contains a mixture of secure (HTTPS) and non-secure (HTTP) content:

This page contains both secure and non-secure items. Do you want to display the non secure items?

When you create a mash-up with a non-secure URL, users may see this warning message, depending on their browser security settings. To suppress this warning in Internet Explorer, follow these steps:

- From the Internet Explorer **Tools** menu, select **Internet Options**.
- Click on the **Security** tab and click on the **Custom Level** button.
- Finally, in the **Miscellaneous** section, set **Display Mixed Content** to **Enable**.

See the following screenshot:



3

Exposing Hacks and Hidden Features

In this chapter, we will cover the following recipes:

- Extending the year range to 2025 on calendar pop ups for all date fields in Salesforce
- Populating a historic year range from 1920 on calendar pop ups for all date fields in Salesforce
- Creating a custom Log A Call button from accounts with prepopulated fields
- Presenting an integrated view of cases on the home page
- Removing the "discover what's new in this release" button from the home page

Introduction

In this curiously-titled chapter, we cover recipes that are intended to deal with some aspects of Salesforce that can only be applied or implemented by using non-supported methods.

Anything that is applied to Salesforce but which is not supported should be considered a risk and should only be used with caution for they constitute a "hack", which is described at `http://en.wikipedia.org/wiki/Hack_%28computer_science%29#In_computer_science` as:

> *In modern computing terminology, a kludge (or often a "hack") is a solution to a problem, doing a task, or fixing a system that [...] (more or less) works.*

[Remember these hidden features or hacks are *not* supported by Salesforce.]

Extending the year range to 2025 on calendar pop ups for all date fields in Salesforce

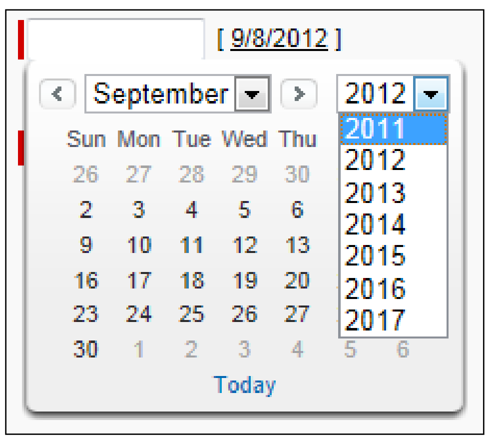
Salesforce.com provides many useful field types from formulae and roll-up summaries to the simple URL field. One field that can be a little frustrating for users, due to the way it allows only a limited selection of year values to be chosen from the pick list, is the date field.

When clicking on a date field, the user is automatically presented with a pop up calendar and when selecting the year pick list, they can only select from a range of seven years. The seven-year range is based on the current year.

The minimum year selectable is the year prior to the current year and the maximum value of year that can be selected is the current year plus five years.

> This is not useful for trying to enter dates further than five years into the future.

For example, when accessing the calendar in 2012, the year range is from 2011 to 2017 as shown in the following screenshot:



There are many business requirements to choose dates further than five years into the future and so this feature poses user experience issues.

Here we are going to improve the user experience and enable future years to be selected.

How to do it...

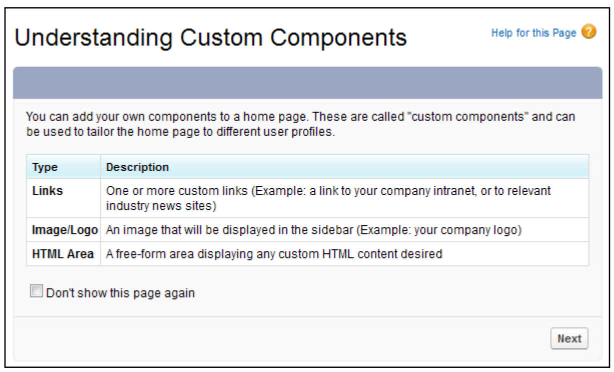
Carry out the following steps to extend the year range on calendar pop ups for all date fields in Salesforce:

1. Navigate to the home page components setup page, by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown).

   The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the checkbox **Don't show this page again** has not been previously checked) as shown in the following screenshot:



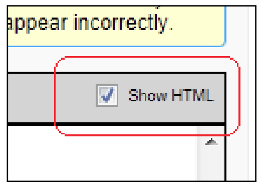
> Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the **Custom Component** in the **Name** field. For this recipe, type the text `Calendar Year Extender`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure the **Narrow (Left) Column** option is selected within the **Component Position** option list.

[Note that you are unable to change this setting after the component is created.]

8. Check the **Show HTML** checkbox.

[The previous step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:



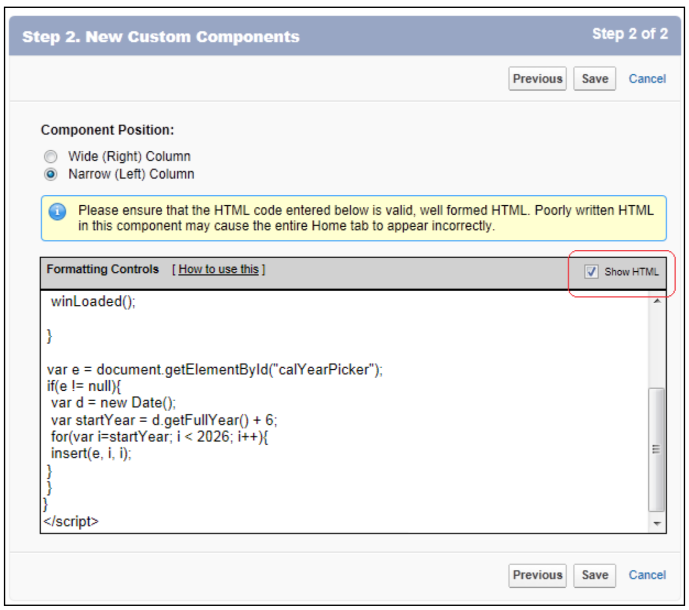
]

9. Paste the following code in the custom component textbox:

```
<br>

<script type="text/javascript">

function insert(pobjSelect, psText, psValue){

 var lobjOption = document.createElement("Option");

lobjOption.text = psText;
 lobjOption.value = psValue;
 pobjSelect.options.add(lobjOption);

}

var winLoaded = window.onload;

window.onload = function(){

if(winLoaded){

  winLoaded();

 }
```

```
 var e = document.getElementById("calYearPicker");
 if(e != null){
  var d = new Date();
  var startYear = d.getFullYear() + 6;
  for(var i=startYear; i < 2026; i++){
  insert(e, i, i);
 }
 }
}
</script>
```

This is shown in the following screenshot:

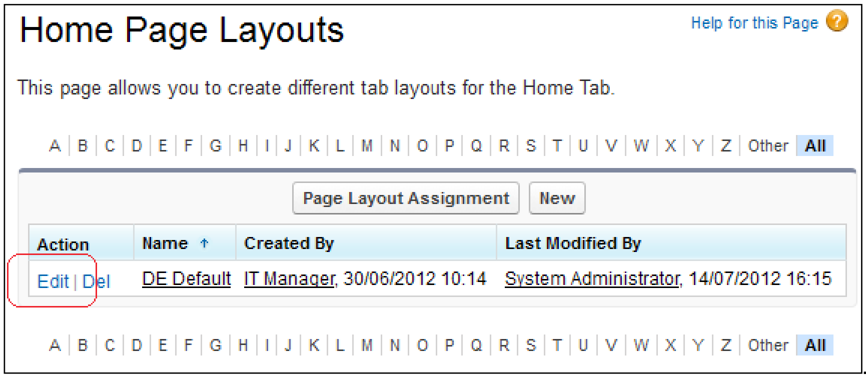


10. Click on **Save**.

> We now need to add the custom home page component to a home page layout.

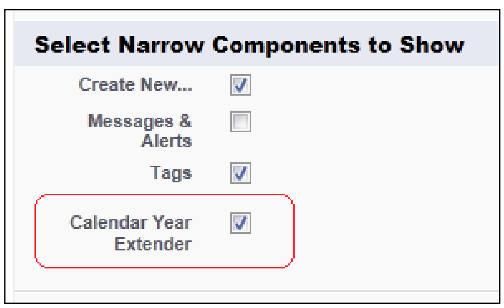
11. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.

12. Determine which home page layout to place the component on and click on **Edit**. Here, in the following screenshot, we are editing the home page layout named **DE Default**:



We are presented with the **Step 1. Select the components to show** page.

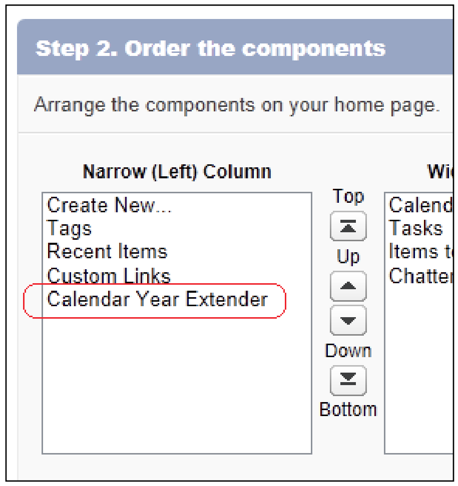
13. Check the **Calendar Year Extender** checkbox in the **Select Narrow Components to Show** section as shown:



14. Click on **Next**.

We are presented with the **Step 2. Order the components** page.

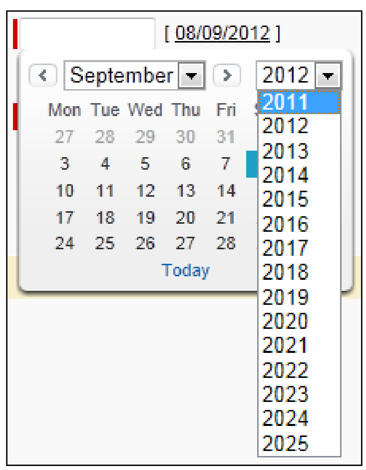
15. Position the **Calendar Year Extender** option at the lowest position in the **Narrow (Left) Column** text area using the **Arrange the components on your home page** section as shown in the following screenshot:



16. Finally, click on **Save**.

How it works...

By using JavaScript, we are able to override the standard year pick list options. You can see what this looks like in the following screenshot:



There's more...

If you wish to shorten or extend the year range, you can modify the highlighted section of the code and replace it with a suitable value of your choice:

```
…
  var startYear = d.getFullYear() + 6;
  for(var i=startYear; i < 2026; i++){
  insert(e, i, i);
…
```

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard), you must ensure that the code is valid.

> Please ensure that the HTML code entered is valid and well formed. Poorly written HTML in this component may cause the entire **Home** tab to appear incorrectly.

There is a maximum of 20 custom components that can be added to a home page layout.

This solution will extend the year range for all pages where you are able to display a sidebar, hence it will not work for pages where there is no sidebar such as the **Reports** tab.

> Remember this hidden feature or hack is *not* supported by Salesforce.

See also

- The *Populating a historic year range from 1920 on calendar pop ups for all date fields in Salesforce* recipe in this chapter

Populating a historic year range from 1920 on calendar pop ups for all date fields in Salesforce

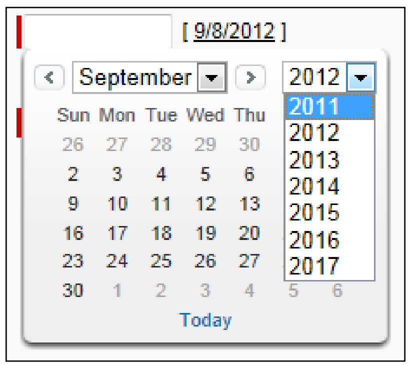
Salesforce.com provides many useful field types from formulae and roll-up summaries to the simple URL field. One field that can be a little frustrating for users, due to the way it only allows a limited selection of year values to be chosen from the pick list, is the date field.

When clicking on a date field, the user is automatically presented with a pop up calendar and when selecting the year pick list, they can only select from a range of seven years. The seven-year range is based on the current year.

The minimum year selectable is the year prior to the current year and the maximum value of year that can be selected is the current year plus five years.

> This is not useful for trying to enter historic dates such as birth dates.

For example, when accessing the calendar in 2012, the year range is from 2011 to 2017 as shown:



There are many business requirements to choose historical dates and so this feature poses user experience issues.

Here we are going to improve the user experience and enable historic years to be selected with this recipe.

How to do it...

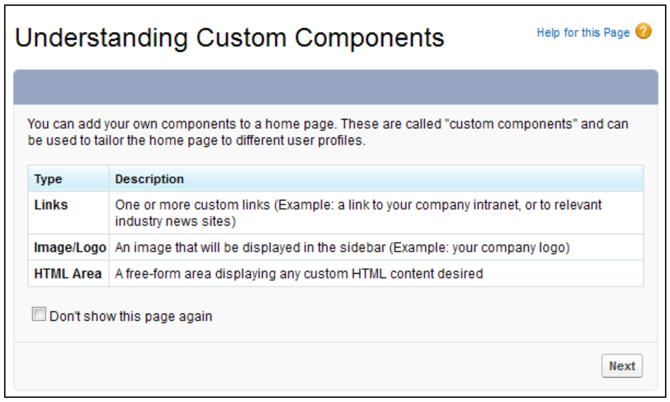
Carry out the following steps to show historic year selections on calendar pop ups for all date fields in Salesforce:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown).

   The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the checkbox **Don't show this page again** has not been previously checked) as in the following screenshot:



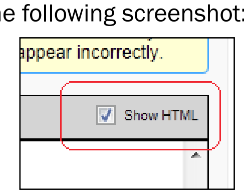
[Here, we are presented with the **Step 1. New Custom Components** page.]

4. Enter the **Custom Component** name in the **Name** field. For this recipe, type the text `Calendar Year Historic`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure the **Narrow (Left) Column** option is selected within the **Component Position** option list.

[Note that you are unable to change this setting after the component is created.]

8. Check the **Show HTML** checkbox.

[The previous step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:



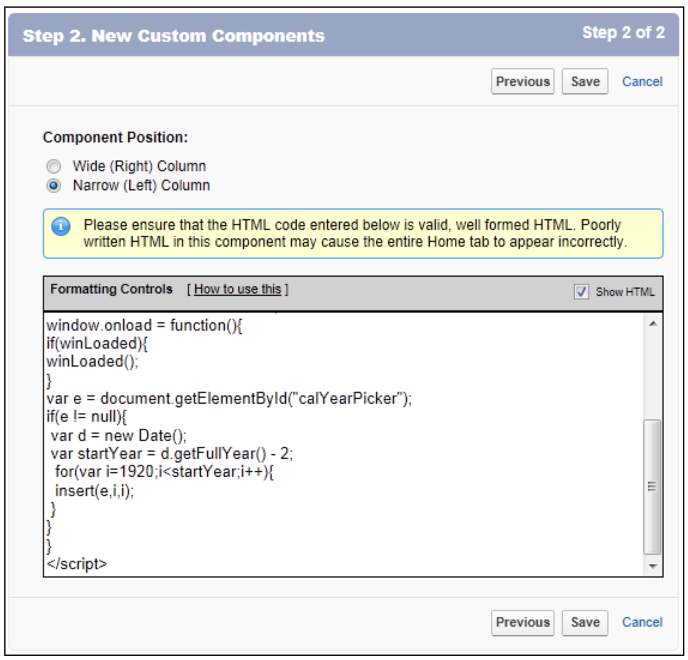
]

9. Paste the following code:

```
<br>

<script type="text/javascript">

function insert(pobjSelect, psText, psValue){

 var lobjOption = document.createElement("Option");

lobjOption.text = psText;
 lobjOption.value = psValue;
 pobjSelect.options.add(lobjOption);

}

var winLoaded = window.onload;

window.onload = function(){

if(winLoaded){

  winLoaded();

 }

 var e = document.getElementById("calYearPicker");

 if(e != null){

  for(i = wavelength - 1; i>=0; i--){

   e.remove(i);
  }
```

```
    var d = new Date();

   var startYear = d.getFullYear() + 6;

    for(var i = 1920; i<startYear; i++){
     insert(e, i, i);
    }
   }

  }

  </script>
```

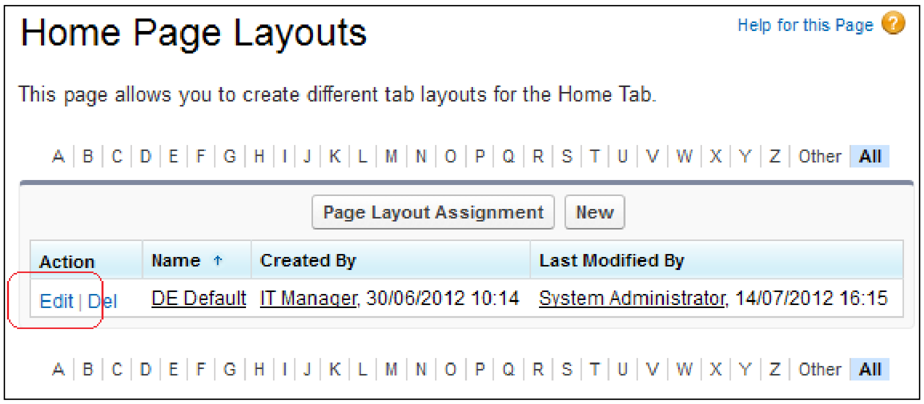
This is shown in the following screenshot:



10. Click on **Save**.

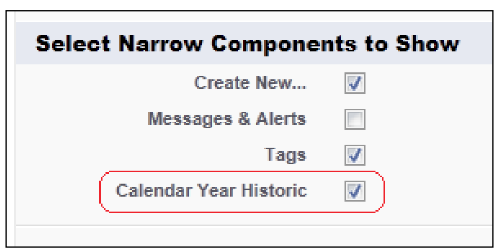
> We now need to add the custom home page component to a home page layout.

11. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here, as shown in the following screenshot, we are editing the home page layout named **DE Default**:



> We are presented with the **Step 1. Select the components to show** page.

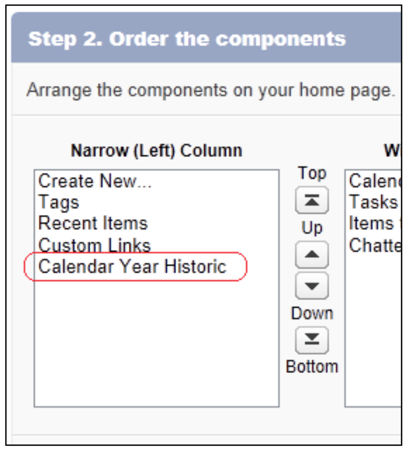
13. Check the **Calendar Year Historic** checkbox in the **Select Narrow Components to Show** section as shown:



14. Click on **Next**.

> We are presented with the **Step 2. Order the components** page.

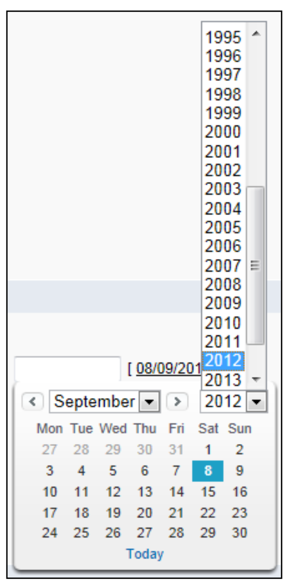
15. Position the **Calendar Year Historic** at the lowest position in the **Narrow (Left) Column** text area using the **Arrange the component on your home page** section as shown:



16. Click on **Save**.

How it works...

By using JavaScript, we are able to override the standard year pick list options. You can see what this looks like in the following screenshot:



There's more...

If you wish to shorten or extend the year range, you can modify the highlighted section of the code and replace it with a suitable value of your choice:

```
…

 var startYear = d.getFullYear() + 6;
  for(var i = 1920; i<startYear; i++){
   insert(e, i, i);
…
```

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard), you must ensure that the code is valid.

> Please ensure that the HTML code entered is valid and well formed. Poorly written HTML in this component may cause the entire **Home** tab to appear incorrectly.

There is a maximum of 20 custom components that can be added to a home page layout.

This solution will show the historic year range for all pages where you are able to display a sidebar, hence it will not work for pages where there is no sidebar such as the **Reports** tab.

> Remember this hidden feature or hack is *not* supported by Salesforce.

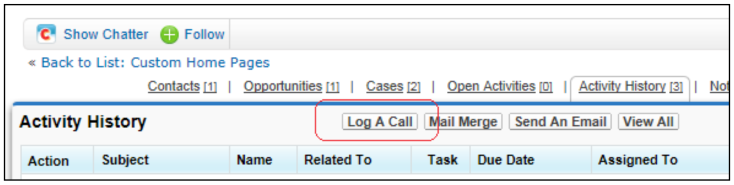
See also

- The *Extending the year range to 2025 on calendar pop ups for all date fields in Salesforce* recipe in this chapter

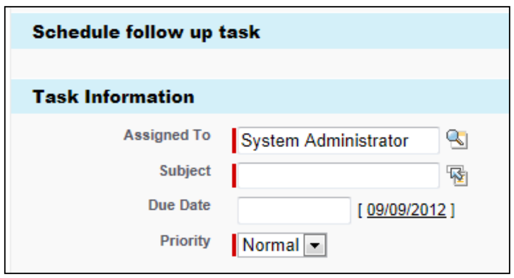
Creating a custom Log A Call button from accounts with prepopulated fields

There is a standard Salesforce **Log A Call** button that allows users to log a call from an account details page which is accessed from the **Activity History** related list.

However, finding and then using this button can be difficult and is not immediately obvious for less experienced users of Salesforce CRM. This requires navigation to the following screen:



This recipe not only allows the option to "log a call" directly from the account details page, but it also allows you to set prepopulated fields on the call task information. In addition, it allows the removal of the follow-up task section (as shown in the following screenshot) which is often considered unnecessary by users:



How to do it...

Using the following steps, we are going to simplify the Log A Call feature with "URL hacking" from a custom button on the account page and improve the user experience:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Buttons and Links**.

   > Scroll down the page to the **Custom Buttons and Links** section.

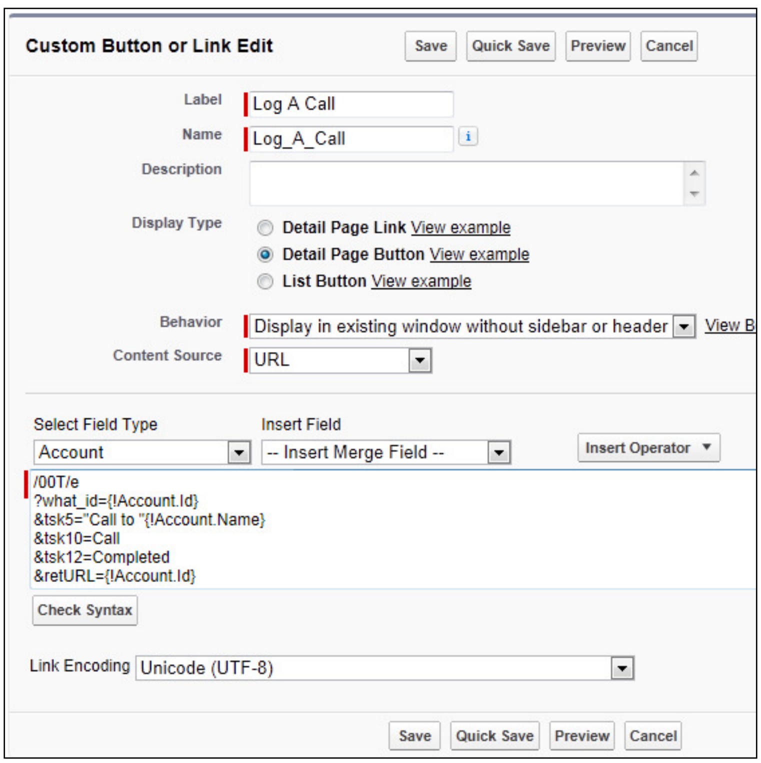
2. Click on **New**.

   > Here, we are presented with the **New Button or Link Edit** page.

3. Enter the label for the button in the **Label** field. For this recipe, type the text `Log A Call`.
4. Accept the default value in the **Name** field which is automatically set as **Log_A_Call**.
5. Choose the **Detail Page Button** option from the **Display Type** options list.
6. Select the **Display in existing window without sidebar or header** selection from the **Behavior** pick list.
7. Accept the default selection of **URL** from the **Content Source** pick list.
8. Paste the following code in the text area given:

```
/00T/e
?what_id={!Account.Id}
&tsk5="Call to "{!Account.Name}
&tsk10=Call
&tsk12=Completed
&retURL={!Account.Id}
```

This is shown in the following screenshot:



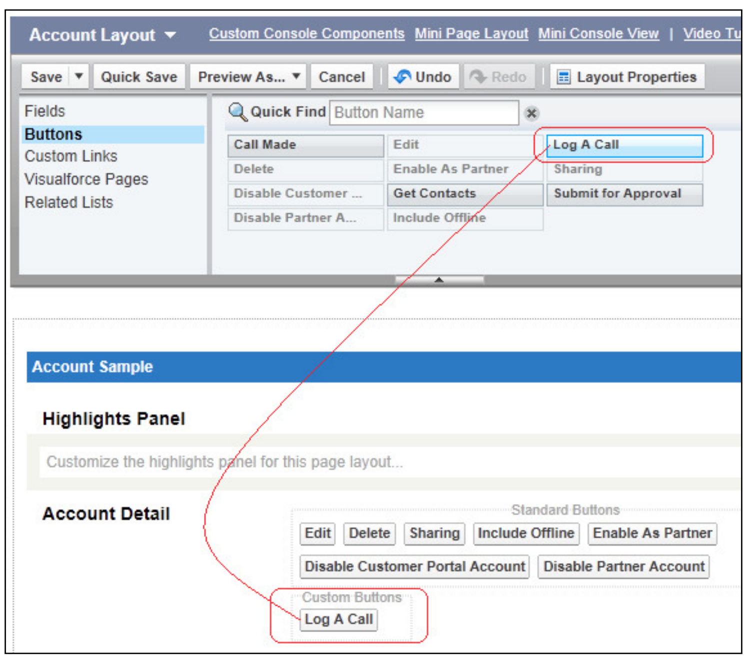
9. Click on **Save**.

> A pop up message appears stating **The new custom button will not be displayed to users until you add it to the page layout**.

10. On the pop up dialog, click on **OK**.
11. Navigate to the account page layout setup page, by following the path, **Your Name** | **Setup** | **Customize** | **Accounts** | **Page Layouts**.
12. Determine which account page layout to place the custom button in and click on **Edit**. Here we are editing the home page layout named **Account Layout**.

> We are presented with the **Account Layout** page.

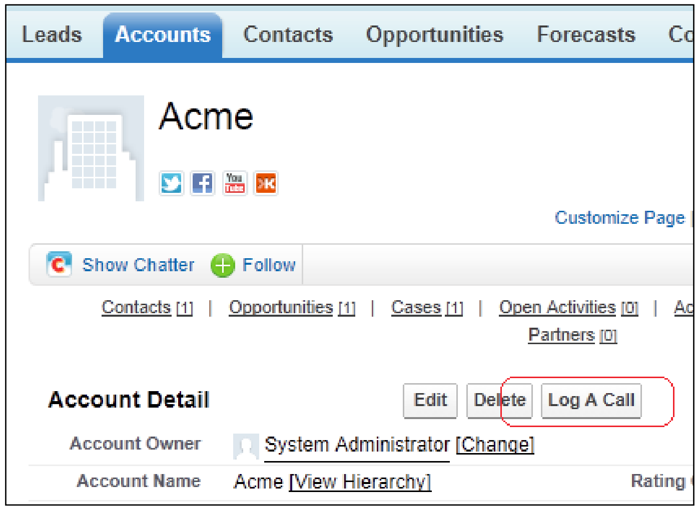
13. Click on the **Buttons** text at the top-left section of the page as shown in the upcoming screenshot.
14. Locate the **Log A Call** button and drag it to the **Custom Buttons** section as shown in the following screenshot:



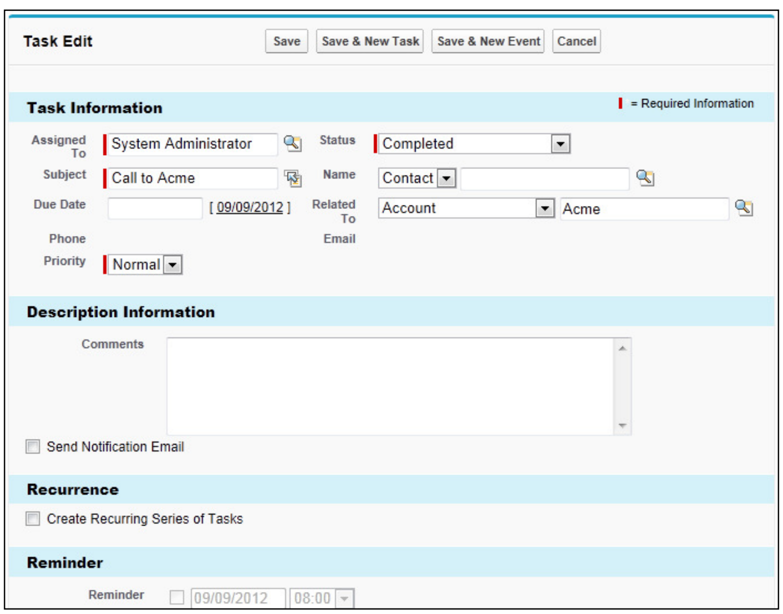
15. Finally, click on **Save**.

How it works...

Clicking on an account record, we are presented with a **Log A Call** custom button as shown:



Upon clicking the **Log A Call** button, the user is presented with a **Task Edit** page with prepopulated values for **Status** and **Subject** as shown:



There's more...

In addition to the prepopulated fields, there is no unnecessary section requiring a follow-up task to be entered.

> Remember this hidden feature or hack is *not* supported by Salesforce.

Presenting an integrated view of cases on the home page

Users often want a simple integrated set of data on their home pages and in particular a way of quickly seeing new cases that have been assigned to them or their team.

For a very quick and easy way of displaying new cases for users on the home page, it is possible to embed the **Case** tab within a home page component using the standard Salesforce **Case** page, but without header or sidebar information.

> Note this approach is not supported by Salesforce and you should consider this use with care as it cannot be relied upon for future use.

To save users' time and showcase information directly on the home page every time they access the **Home** tab, create this custom home page component with this recipe.

How to do it...

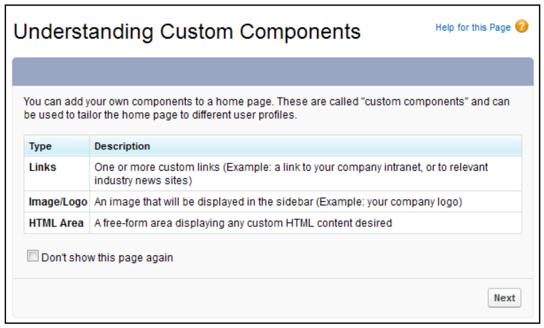
Carry out the following steps to present an integrated view of cases on the home page:

1. Navigate to the home page components setup page by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown).

The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the checkbox **Don't show this page again** has not been checked previously) as in the following screenshot:

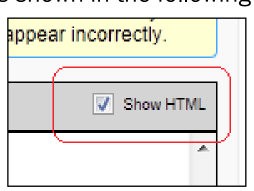


> Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the **Custom Component** in the **Name** field. For this recipe, type the text `Cases Section`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure the **Wide (Right) Column** option is selected within the **Component Position** option list.

> Note that you are unable to change this setting after the component is created.

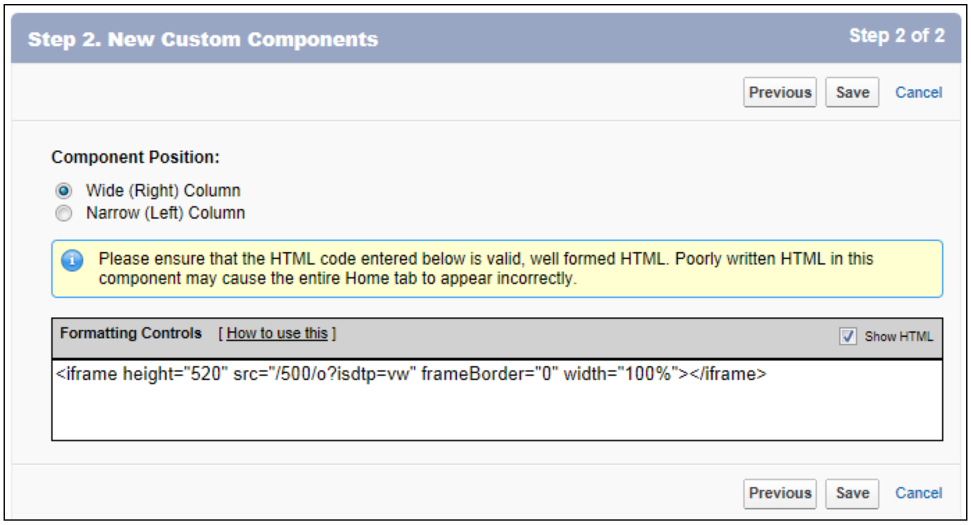
8. Check the **Show HTML** checkbox.

> The previous step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:
>
> 
>

9. Paste the following code in the text area:

```
<iframe height="520" src="/500/o?isdtp=vw" frameBorder="0"
width="100%"></iframe>
```

This is shown in the following screenshot:



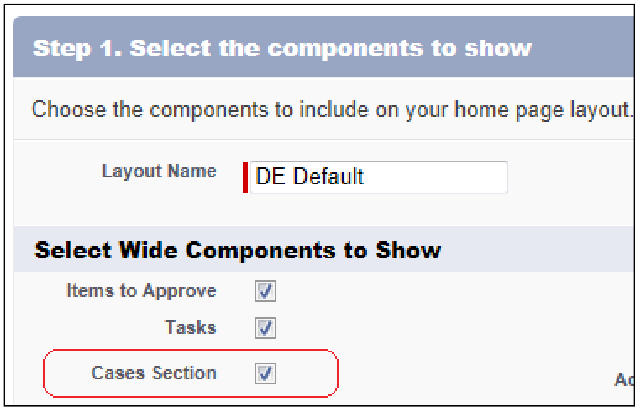
10. Click on **Save**.

> We now need to add the custom home page component to a home page layout.

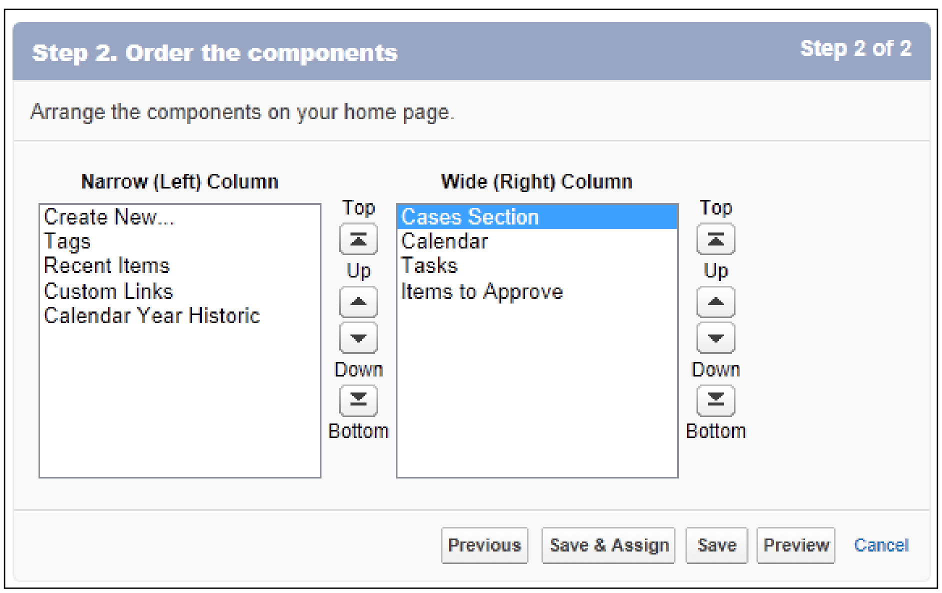
11. Navigate to the home page components setup page, by following the path, **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**:

> We are presented with the **Step 1. Select the components to show** page.

13. Check the **Cases Section** checkbox in the **Select Wide Components to Show** section as shown:



14. Click on **Next**.
15. Position **Cases Section** at the top position in the **Wide (Right) Column** text area using the **Arrange the component on your home page** section as shown:



16. Click on **Save**.

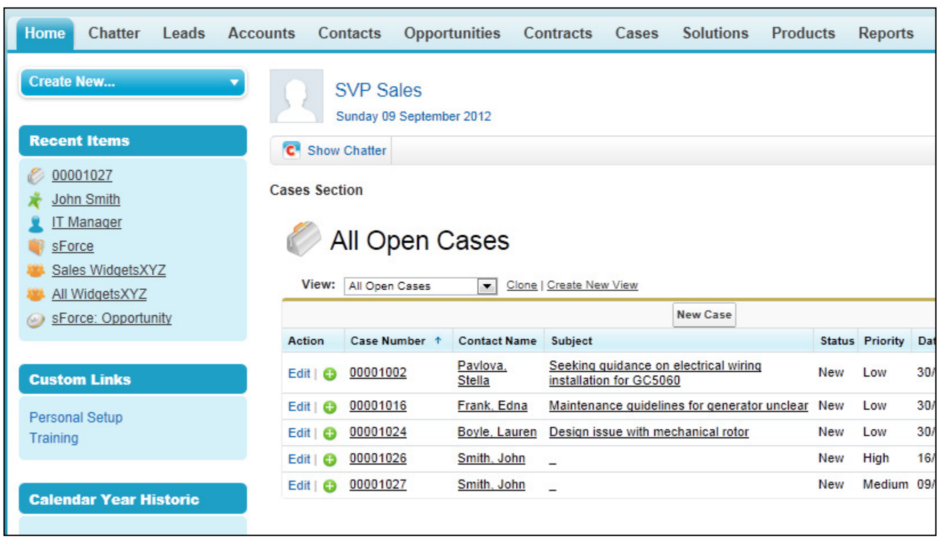
How it works...

This recipe makes use of HTML to create an iframe which is a way of embedding other HTML pages or sections of HTML code either from within Salesforce or pulled from a server external to Salesforce.

Here the location we are using is the standard **Case** tab but with the additional parameter `isdtp`. This parameter is a Salesforce-specific parameter that presents the content without a header or sidebar.

The `isdtp` parameter is typically used behind the scenes to render the pages that are displayed on the Salesforce **Service Cloud** and **Console** tabs.

You can see what this looks like when the **Home** tab is selected as shown in the following screenshot:



There's more...

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard), you must ensure that the code is valid.

> Please ensure that the HTML code entered is valid and well formed. Poorly written HTML in this component may cause the entire **Home** tab to appear incorrectly.

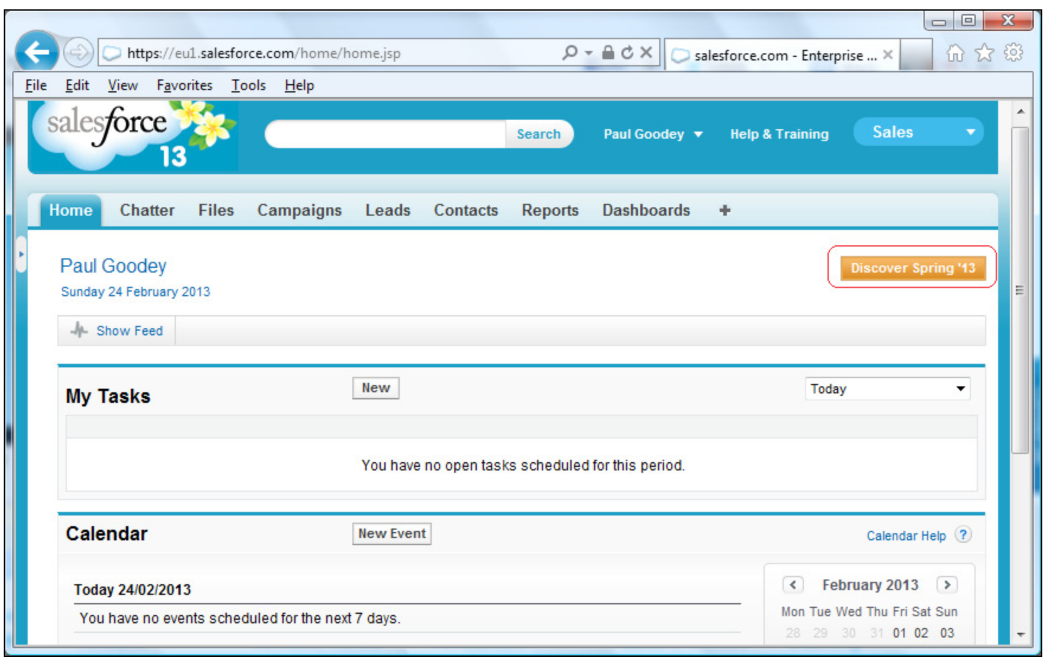
There is a maximum of 20 custom components that can be added to a home page layout.

> Remember this hidden feature or hack is *not* supported by Salesforce.

Removing the "discover what's new in this release" button from the home page

There is a "discover what's new in this release" button on the home page shown by default for all users regardless of their profile in Salesforce. The button provides a link to a web page in Salesforce that contains information and various resources for the new features and changes that have been introduced in the latest release. However, since there is no option to hide this button, it can sometimes be a little confusing or irritating for some users.

The button appears with the following text: **Discover Spring '13** (**Spring '13** was the latest Salesforce CRM release at the time of writing), and is presented at the top-right corner of the home page as shown in the following screenshot:



How to do it...

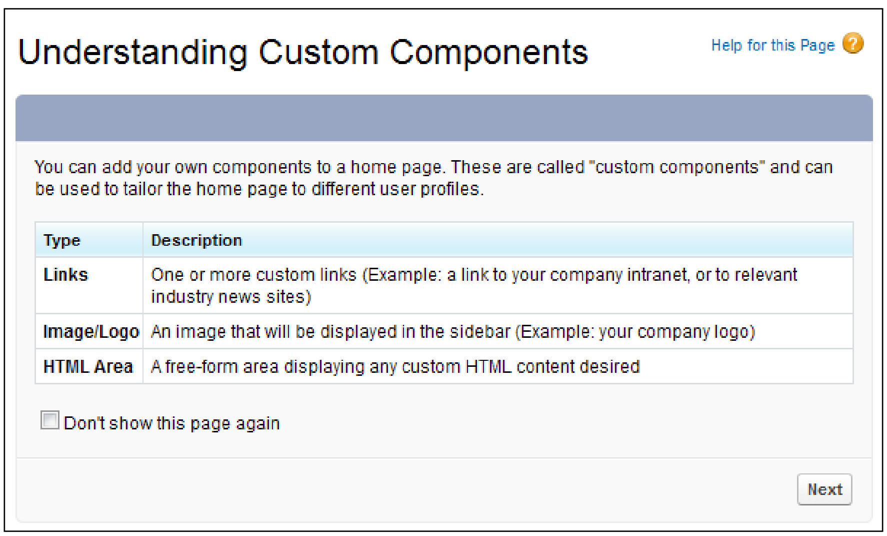
Carry out the following steps to remove the "discover what's new in this release" button from the home page:

1. Navigate to the home page components setup page, by going to **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Components**.
2. Click on **New**.

> The **New** button is found by scrolling down the page to the **Custom Components** section.

3. Click on **Next** (on the **Understanding Custom Components** splash screen if shown).

   The **Next** button is found on the **Understanding Custom Components** splash screen (this page is only shown if the checkbox **Don't show this page again** has not been checked previously) as in the following screenshot:



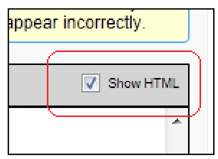
> Here, we are presented with the **Step 1. New Custom Components** page.

4. Enter the name of the **Custom Component** in the **Name** field. For this recipe, type the text `Hide Whats New Button`.
5. Select the **HTML Area** option from the **Type** options list.
6. Click on **Next**.
7. Ensure the **Wide (Right) Column** option is selected within the **Component Position** option list.

> Note that you are unable to change this setting after the component is created.

8. Check the **Show HTML** checkbox.

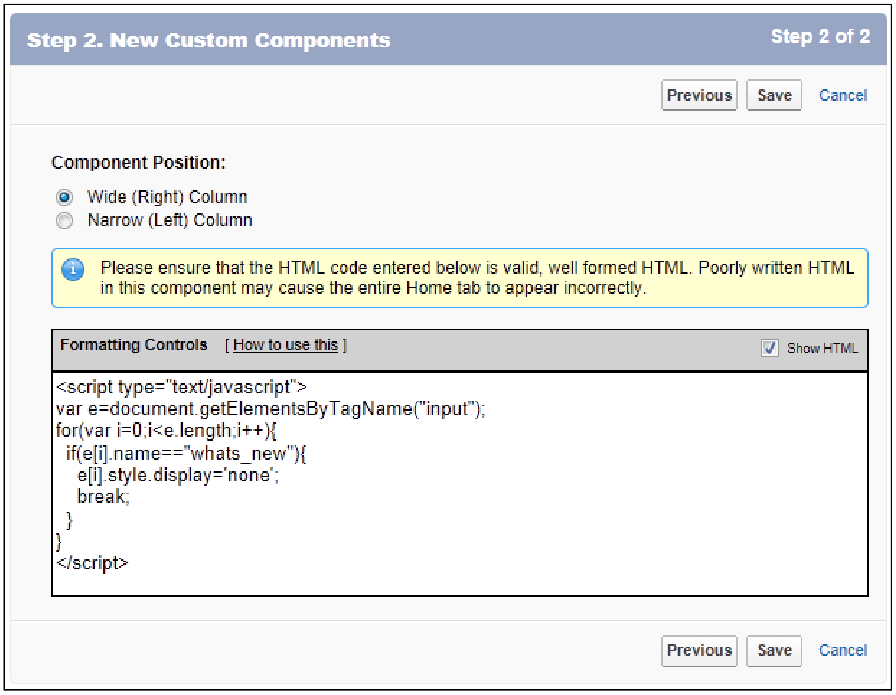
> The previous step is important! Locate and check the **Show HTML** checkbox as shown in the following screenshot:



9. Paste the following code in the text area:

```
<script type="text/javascript">
var e=document.getElementsByTagName("input");
for(var i=0;i<e.length;i++){
  if(e[i].name=="whats_new"){
    e[i].style.display='none';

    break;
  }
}
</script>
```

This is shown in the following screenshot:



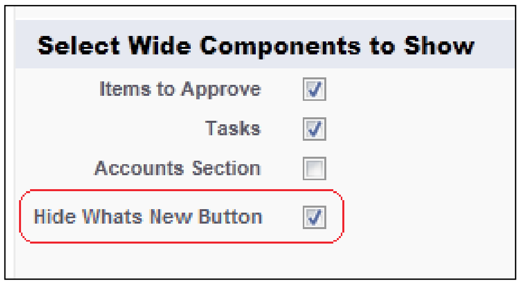
10. Click on **Save**.

> We now need to add the custom home page component to a home page layout.

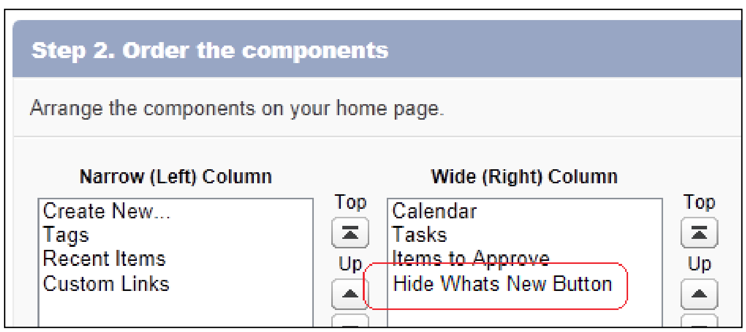
11. Navigate to the home page components setup page, by following the path, **Your Name** | **Setup** | **Customize** | **Home** | **Home Page Layouts**.
12. Determine which home page layout to place the component on and click on **Edit**. Here we are editing the home page layout named **DE Default**:

We are presented with the **Step 1. Select the components to show** page.

13. Check the **Hide Whats New Button** checkbox in the **Select Wide Components to Show** section as shown in the following screenshot:



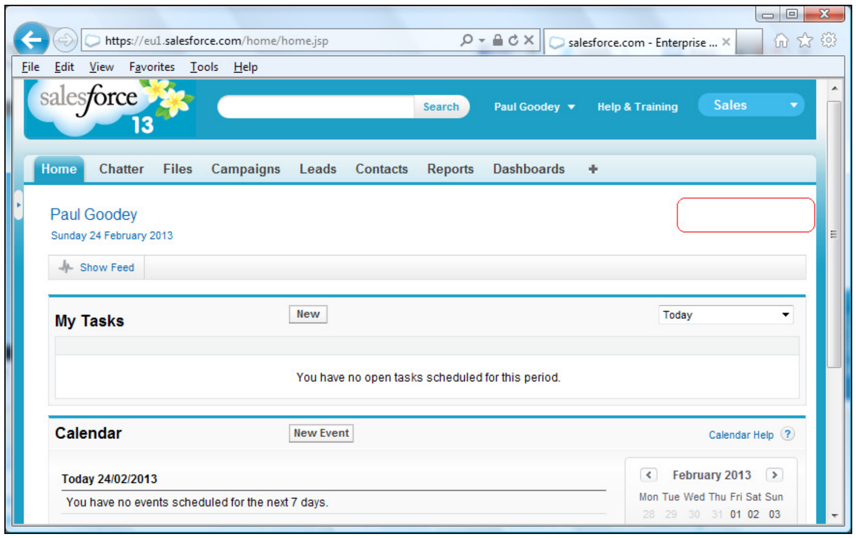
14. Click on **Next**.
15. Position the **Hide Developer Button** at the top position in the **Wide (Right) Column** text area using the **Arrange the component on your home page.** section as shown in the following screenshot:



16. Finally, click on **Save**.

How it works...

You can see that the "discover what's new in this release" button is now removed and what this looks like when the **Home** tab is selected as shown in the following screenshot:



There's more...

When entering HTML and JavaScript code into the HTML editor section (in step 2 of the **New Custom Component** wizard), you must ensure that the code is valid.

> Please ensure that the HTML code entered is valid and well formed. Poorly written HTML in this component may cause the entire **Home** tab to appear incorrectly.

There is a maximum of 20 custom components that can be added to a home page layout.

> Remember this hidden feature or hack is *not* supported by Salesforce.

See also

- The *Removing chatter feeds from the home page* recipe in *Chapter 1, Working with Home Page Components and Custom Links*

4
Automating Salesforce CRM

In this chapter, we will cover the following recipes:

- Deriving year and month values from the opportunity close date using a formula
- Calculating the week number for the opportunity close date using a formula
- Creating an opportunity close date e-mail alert using workflow
- Setting a default opportunity name using a field update workflow
- Generating a default opportunity close date using a field update workflow

Introduction

A CRM system must be as productive as possible to justify its investment, hence, if there are any aspects that can be configured and set up to be used more efficiently, they are usually worth exploring.

The Salesforce CRM application aims to be as efficient as possible out of the box, however, typically there are company-specific business rules and process flows that need to be implemented.

This is where the power of the Salesforce CRM application and the Force.com platform becomes truly apparent.

Within the Salesforce CRM application and the Force.com platform, you can easily create workflow to trigger your company's unique business rules and logic.

Salesforce CRM also enables you to create complex formulae to generate data records as well as extend the user interface to create custom buttons that prepopulate data fields.

Deriving year and month values from the opportunity close date using a formula

To simplify the format of dates for presentation and reporting, we can derive the year and month from the date element.

In this recipe, we will display a derived year and month text string for the opportunity close date, on the opportunity record detail page, calculated from the standard date field called `CloseDate`.

How to do it...

Carry out the following steps to derive the year and month values from the opportunity close date using a formula:

1. Navigate to the opportunity customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Fields**.

   > Scroll down to the opportunity **Custom Fields & Relationships** section.

2. Click on **New**.

   > We are presented with the **Step 1. Choose the field type** page.

3. Select the **Formula** option.
4. Click on **Next**.

   > We are presented with the **Step 2. Choose output type** page.

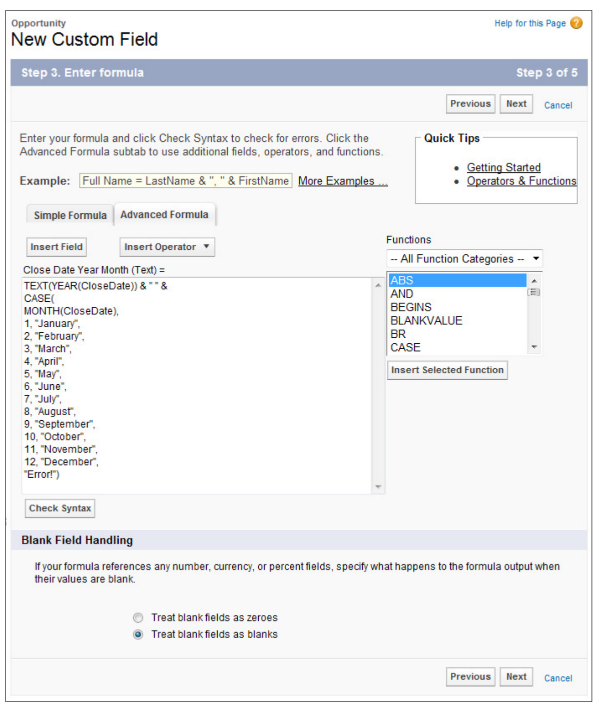
5. Type `Close Date Year Month` in the **Field Label** textbox.
6. Click on the **Field Name**. When clicking out of the **Field Label** textbox the **Field Name** is automatically filled with the value **Close_Date_Year_Month**.
7. Set the formula return type as **Text**.
8. Click on **Next**.

[We are presented with the **Step 3. Enter formula** page.]

9. Paste the following code in the formula editor box:

```
TEXT(YEAR(CloseDate)) & " " &
CASE(
MONTH(CloseDate),
1, "January",
2, "February",
3, "March",
4, "April",
5, "May",
6, "June",
7, "July",
8, "August",
9, "September",
10, "October",
11, "November",
12, "December", "Error!")
```

This is shown in the following screenshot:



10. In the **Blank Field Handling** section, select the option **Treat blank fields as blanks**.
11. Click on **Next**.

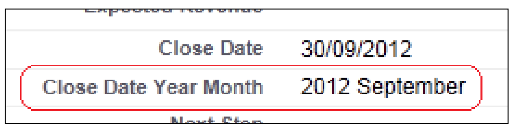
> We are presented with the **Step 4. Establish field-level security** page.

12. Select the profiles to which you want to grant read access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
13. Finally, click on **Save**.

How it works...

The formula field appears on the opportunity detail page.

You can see what this looks like in the following image:



Calculating the week number for the opportunity close date using a formula

In this recipe, we will display the week number according to the opportunity close date on the opportunity record detail page, where the week number is calculated from the standard date field called `CloseDate`.

How to do it...

Carry out the following steps to calculate the week number for the opportunity close date using a formula:

1. Navigate to the opportunity customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Fields**.

> Scroll down to the **Opportunity Custom Fields & Relationships** section.

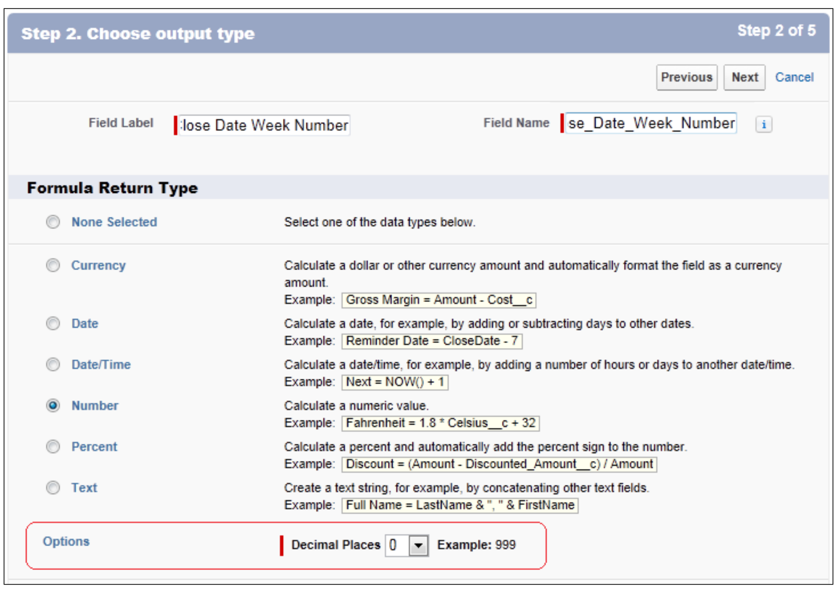
2. Click on **New**.

[We are presented with the **Step 1. Choose the field type** page.]

3. Select the **Formula** option.
4. Click on **Next**.

[We are presented with the **Step 2. Choose output type** page.]

5. Type `Close Date Week Number` in the **Field Label** textbox.
6. Click on the **Field Name** textbox. When clicking out of the **Field Label** textbox, the **Field Name** textbox is automatically filled with the value **Close_Date_Week_Number**.
7. Set **Formula Return Type** as **Number**.
8. Set **Decimal Places** to **0** in the **Options** section as shown in the following screenshot:



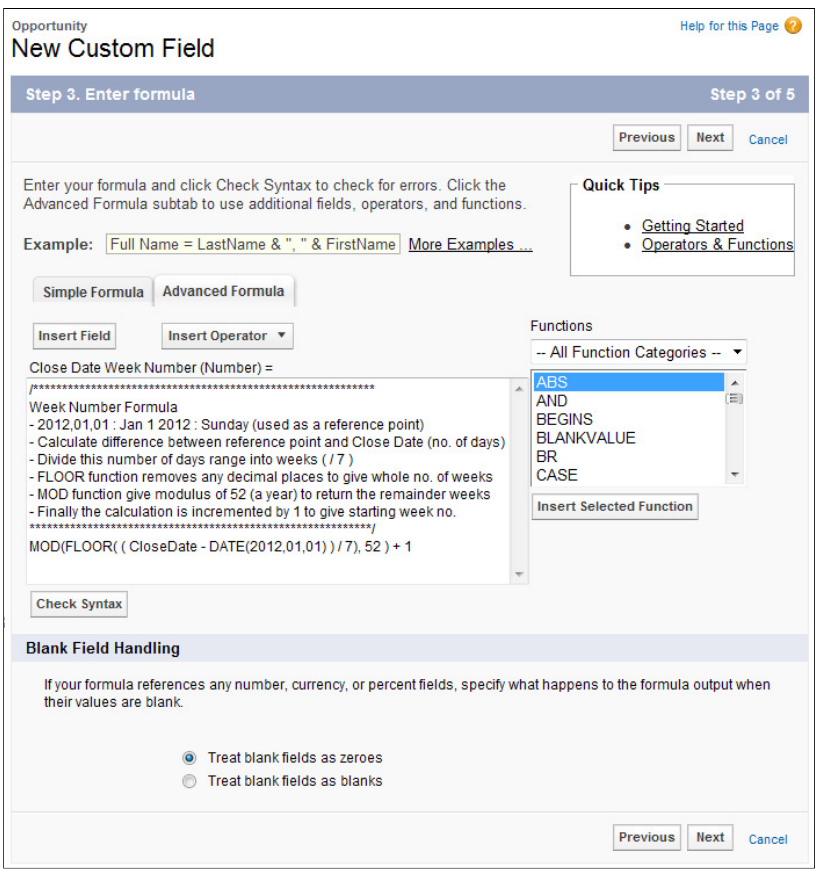
9. Click on **Next**.

[We are presented with the **Step 3. Enter formula** page.]

10. Paste the following code in the formula editor box:

```
/************************************************************
Week Number Formula
- 2012,01,01 : Jan 1 2012 : Sunday (used as a reference point)
- Calculate difference between reference point and Close Date (no.
of days)
- Divide this number of days range into weeks ( / 7 )
- FLOOR function removes any decimal places to give whole no. of
weeks
- MOD function give modulus of 52 (a year) to return the remainder
weeks
- Finally the calculation is incremented by 1 to give starting
week no.
************************************************************/
MOD(FLOOR( ( CloseDate - DATE(2012,01,01) ) / 7), 52 ) + 1
```

This is shown in the following screenshot:



11. In the **Blank Field Handling** section, select the option **Treat blank fields as zeroes**.
12. Click on **Next**.

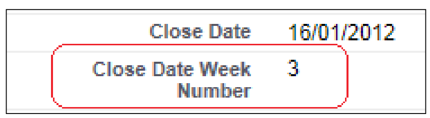
> We are presented with the **Step 4. Establish field-level security** page.

13. Select the profiles to which you want to grant read access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
14. Finally, click on **Save**.

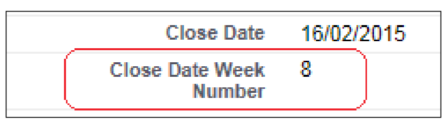
How it works...

The formula field appears on the opportunity detail page.

As an example, you can see how this appears as week **3** for a `CloseDate` value of **16/01/2012** as in the following screenshot:



Another example shows how this appears as week **8** for a `CloseDate` value of **16/02/2015** as in the following screenshot:



Creating an opportunity close date e-mail alert using workflow

As best practice in CRM systems, when creating sales opportunities there should always be a close date with which we can forecast, report, and monitor the progress of the sale.

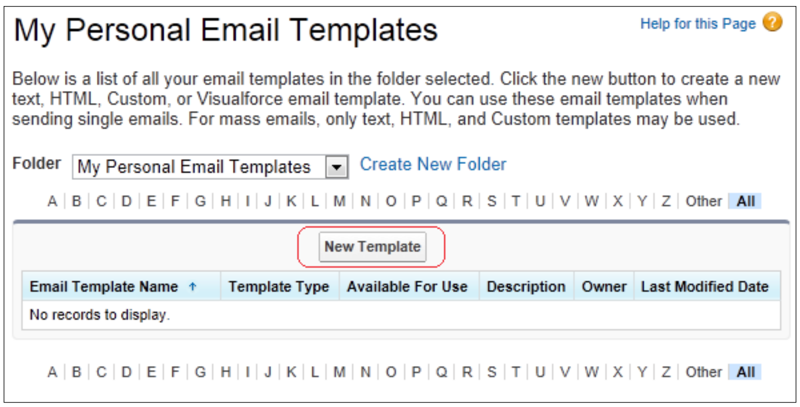
In Salesforce CRM, the close date can be referred to in order to automatically trigger an e-mail that alerts the opportunity owner that the sales opportunity is nearing the scheduled close date.

To create this automatic sending of close date e-mail alerts, the following recipe makes use of a Salesforce workflow rule and e-mail alert.

Getting ready

Create an e-mail template using the following steps:

1. Navigate to the new e-mail template creation page, by going to **Your Name** | **Setup** | **Email** | **My Templates**.
2. Select **My Personal Email Templates** from the **Folder** drop-down list.
3. Click on the **New Template** button as shown in the following screenshot:



> Here, we are presented with the **Step 1: Email Template: New Template** page.

4. Choose the **Text** option and click on the **Next** button.

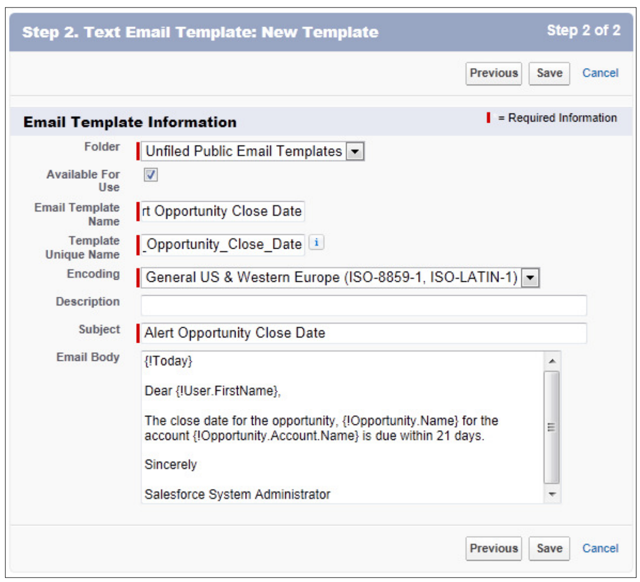
> Here, we are presented with the **Step 2: Text Email Template: New Template** page.

5. Choose the **Unfiled Public Email Templates** option from the **Folder** drop-down list.
6. Set the checkbox **Available For Use**.
7. Enter the name of the e-mail template in the **Email Template Name** field. In this recipe, type the text `Alert Opportunity Close Date`.
8. Accept the default text `Alert_Opportunity_Close_Date` that is automatically set in the **Template Unique Name** field.
9. Optionally, enter a description in the **Description** field.
10. Enter the subject for the e-mail in the **Subject** field. In this recipe, type the text `Alert Opportunity Close Date`.

11. In the **Email Body** text area, enter the following code:

```
{!Today}

Dear {!User.FirstName},

The close date for the opportunity, {!Opportunity.Name} for the
account {!Opportunity.Account.Name} is due within 21 days.

Sincerely

Salesforce System Administrator
```

This is shown in the following screenshot:



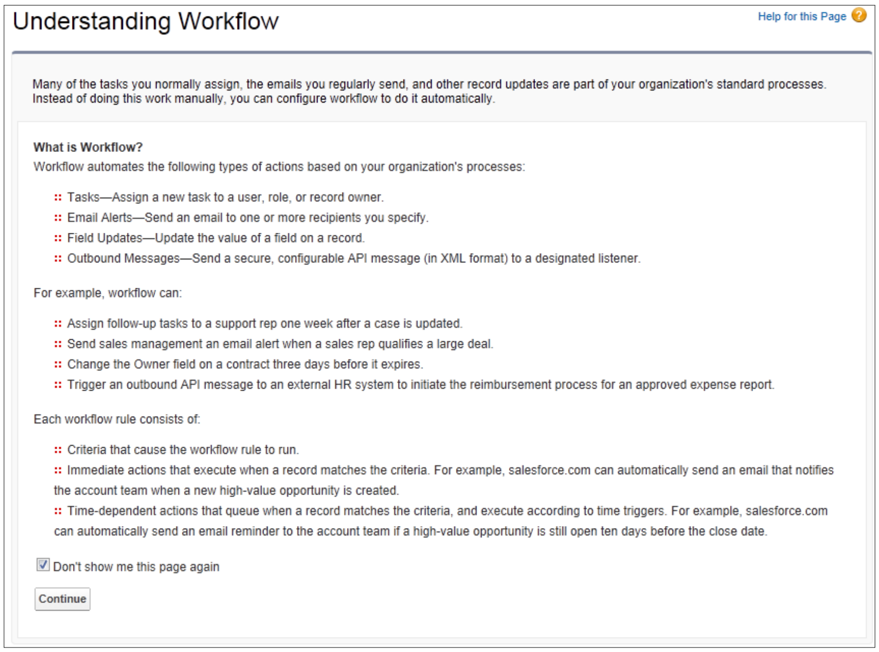
12. Click on **Save**.

How to do it...

Carry out the following steps to create an opportunity close date e-mail alert using workflow:

1. Navigate to the workflow setup page, by going **Your Name** | **Setup** | **Create** | **Workflow & Approvals** | **Workflow Rules**.
2. Click on **Continue** (if shown).

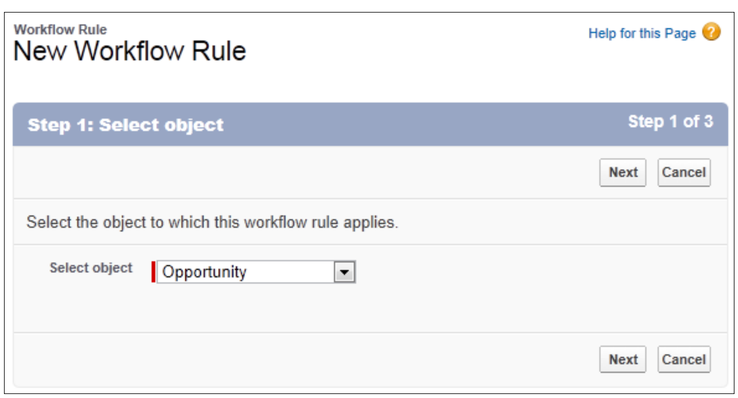
The **Continue** button is found on the **Understanding Workflow** page (if this is shown) as in the following screenshot:



3. Click on **New Rule**.

> Here, we are presented with the **Step 1: Select object** page.

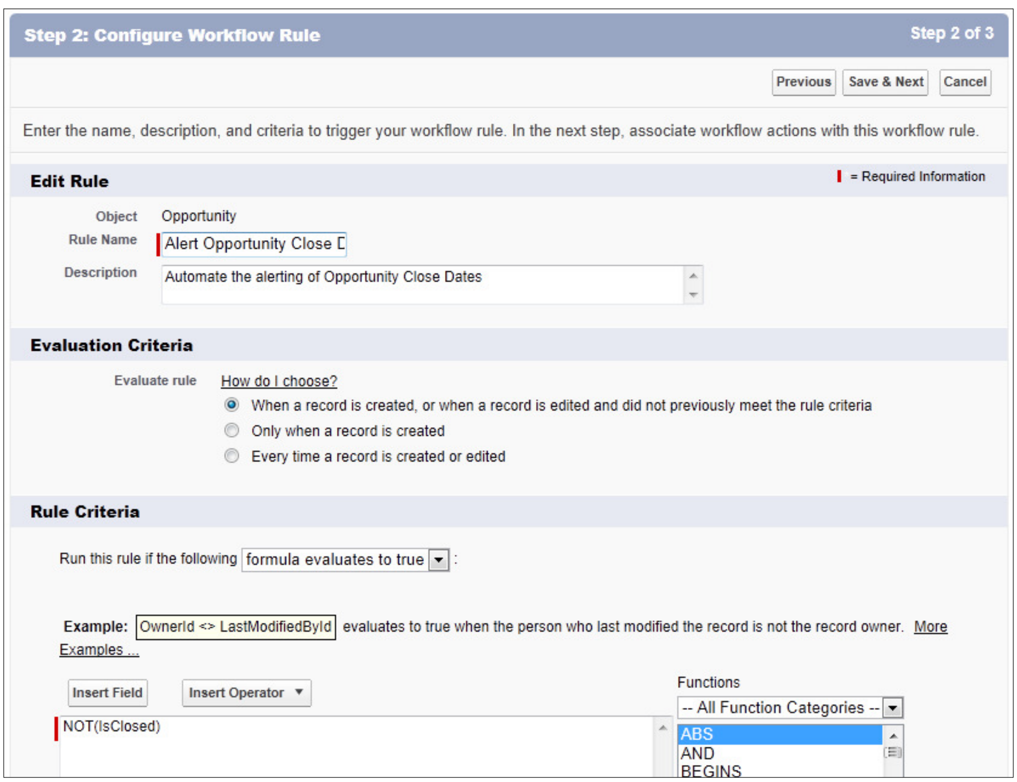
4. Select **Opportunity** and click on **Next** as shown in the following screenshot:



> Here, we are presented with the **Step 2: Configure Workflow Rule** page.

5. Enter the name of the new workflow rule in the **Rule Name** field. For this recipe, type the text `Alert Opportunity Close Date`.
6. Optionally, enter a description for the new workflow rule in the **Description** field. For this recipe, type the text `Automate the alerting of Opportunity Close Dates`.
7. Select the **When a record is created, or when a record is edited and did not previously meet the rule criteria** option from the **Evaluation Criteria** options list.
8. In the **Rule Criteria** section, set the **Run this rule if the following** option to **formula evaluates to true**.
9. Enter the following formula (as shown in the following screenshot):

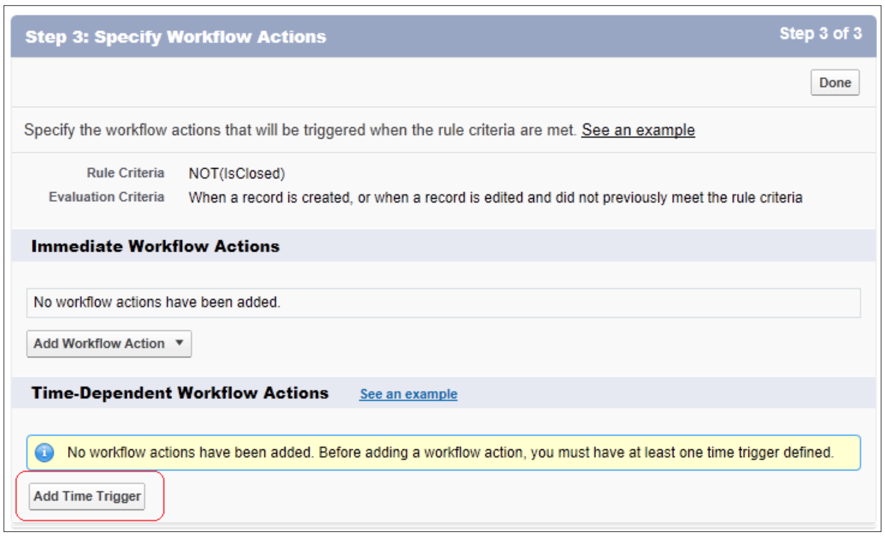
```
NOT(IsClosed)
```



10. Click on **Save & Next**.

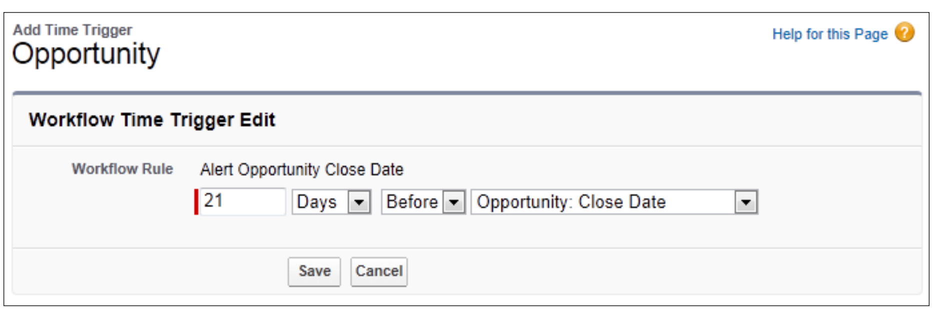
> Here, we are presented with the **Step 3: Specify Workflow Actions** page.

11. In the **Time-Dependent Workflow Actions** section, click on the **Add Time Trigger** button as shown in the following screenshot:



> Here, we are presented with the **Add Time Trigger** page.

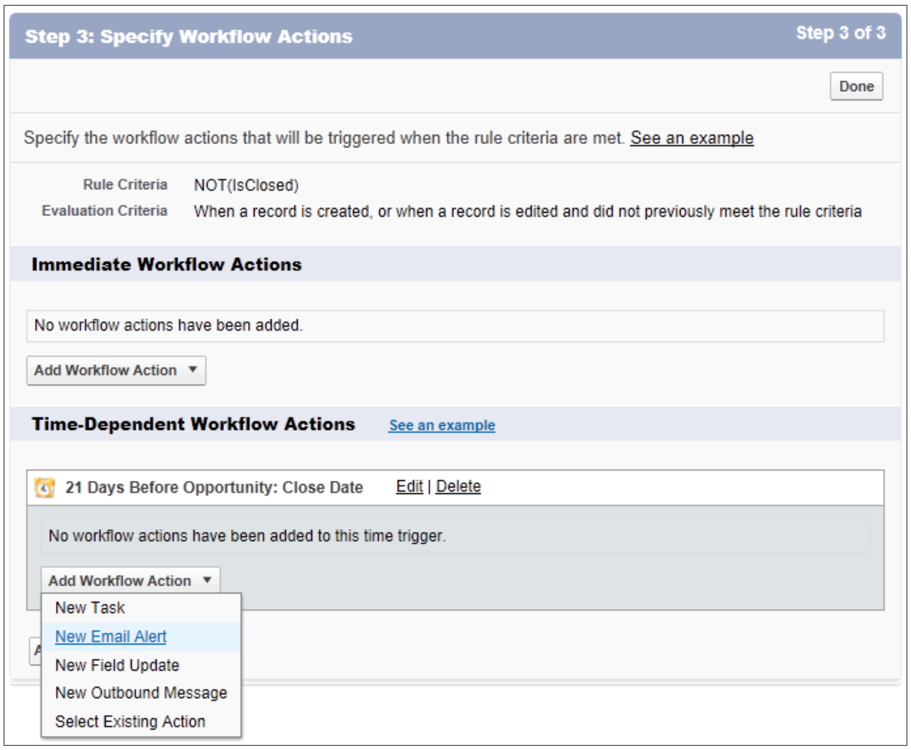
12. Enter **21** in the first text field and select **Days** from the next drop-down list. Select **Before** and **Opportunity Close Date** as shown in the following screenshot:



13. Click on **Save**.

Here, we are presented with the **Step 3: Specify Workflow Actions** page.

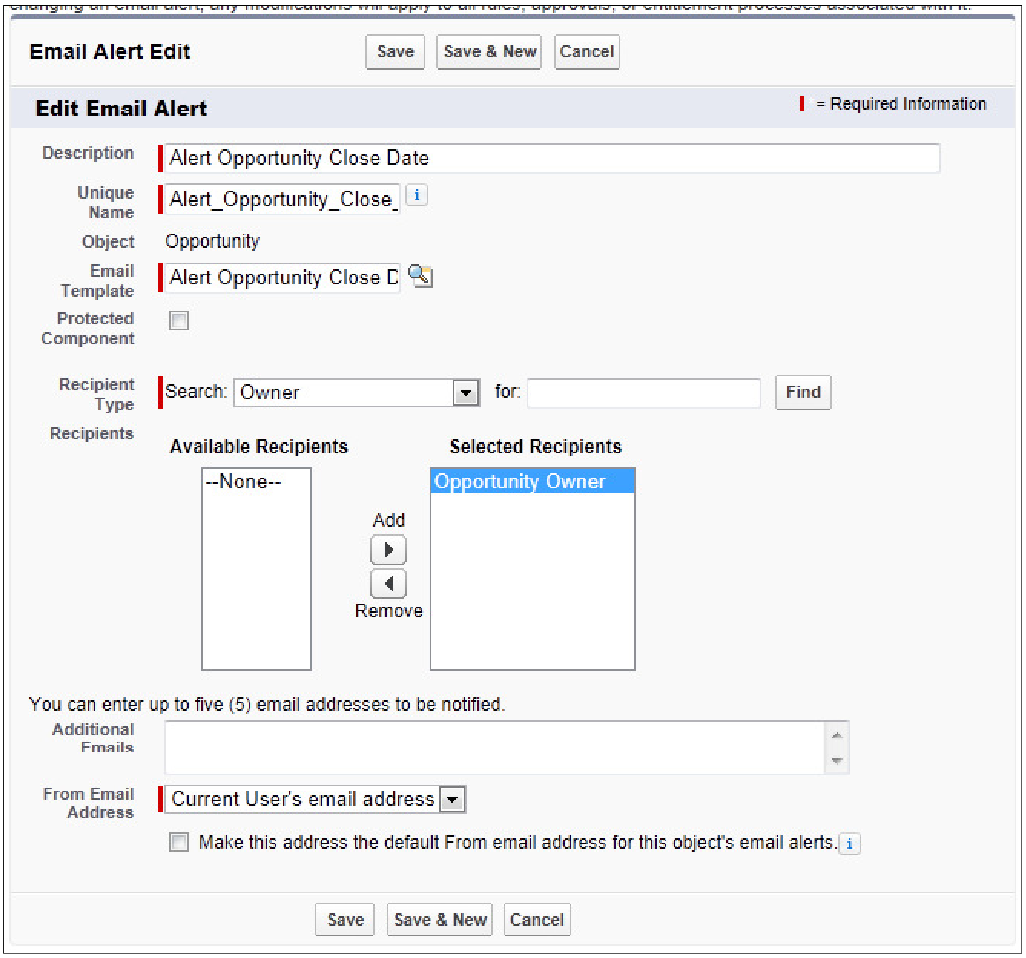
14. In the **Time-Dependent Workflow Actions** section, set the **Add Workflow Action** option to **New Email Alert** as shown in the following screenshot:



Here, we are presented with the **Email Alert Edit** page.

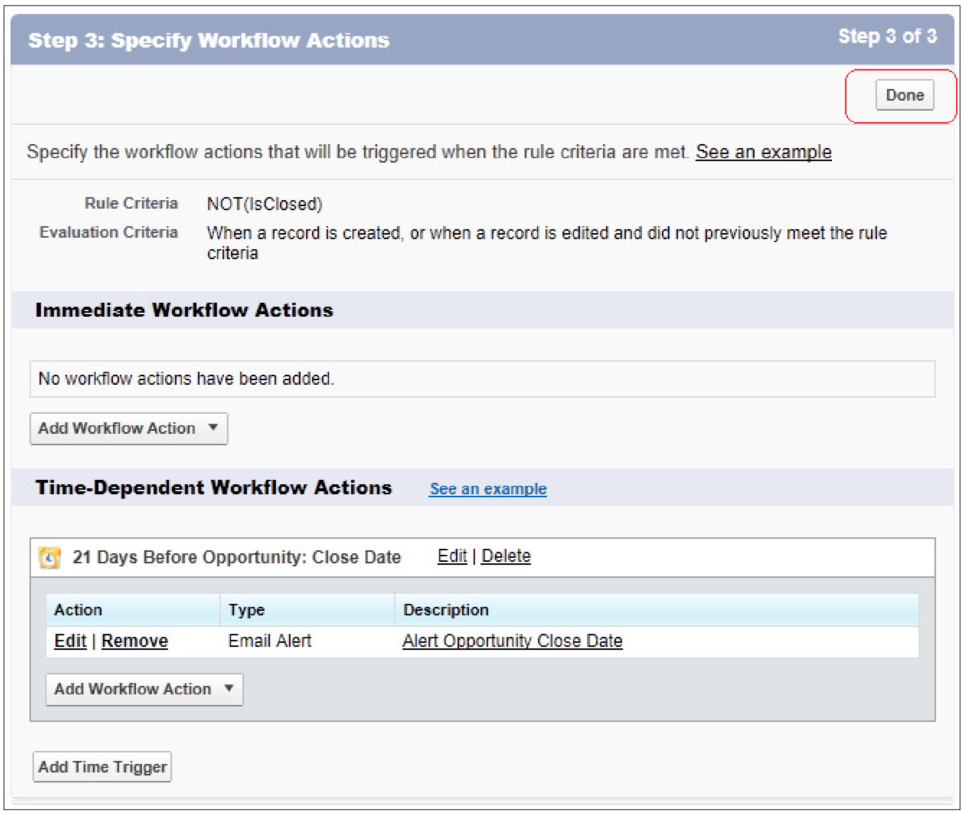
15. Enter the description for the e-mail alert in the **Description** field. For this recipe, type the text `Alert Opportunity Close Date`.
16. Accept the default text **Alert_Opportunity_Close_Date** that is automatically set in the **Unique Name** field.
17. Now choose the template that we created in the *Getting ready* section of this recipe. We called this template **Alert Opportunity Close Date**.
18. Set the **Recipient Type to Owner**.

19. Set the **Selected Recipients** to **Opportunity Owner** as shown in the following screenshot:



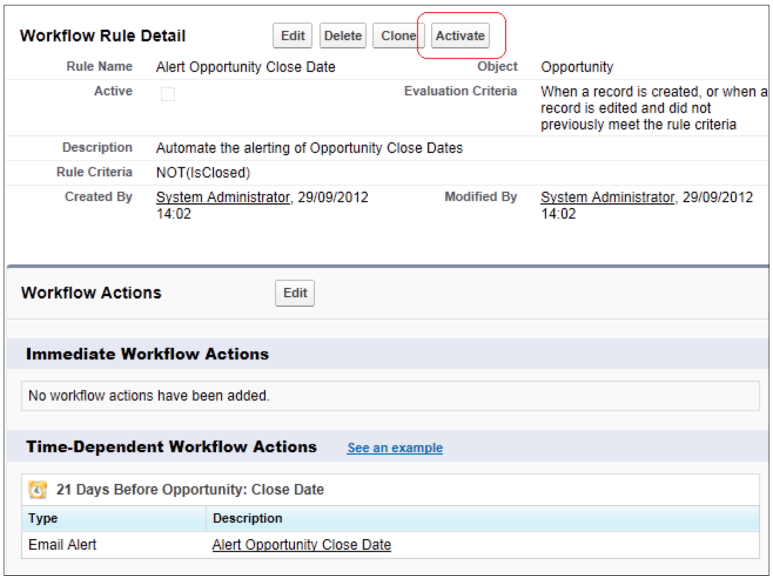
20. Click on **Save**.

> Here, we are presented with the **Step 3: Specify Workflow Actions** page (as shown in the following screenshot).

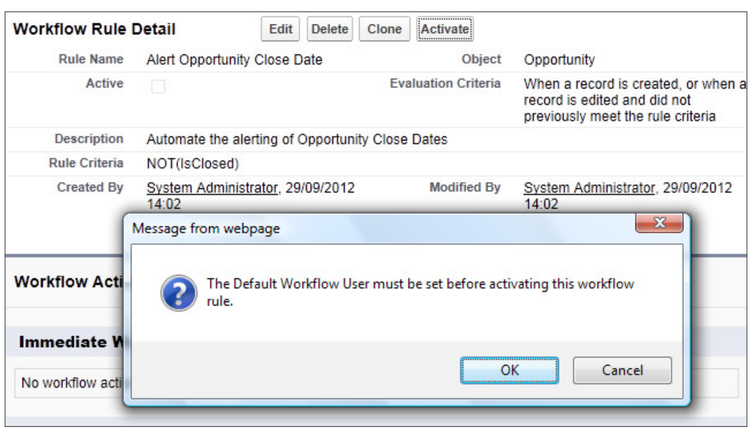


21. Click on **Done**.

> We are now redirected to the new workflow rule where we can activate the rule (as shown in the following screenshot).

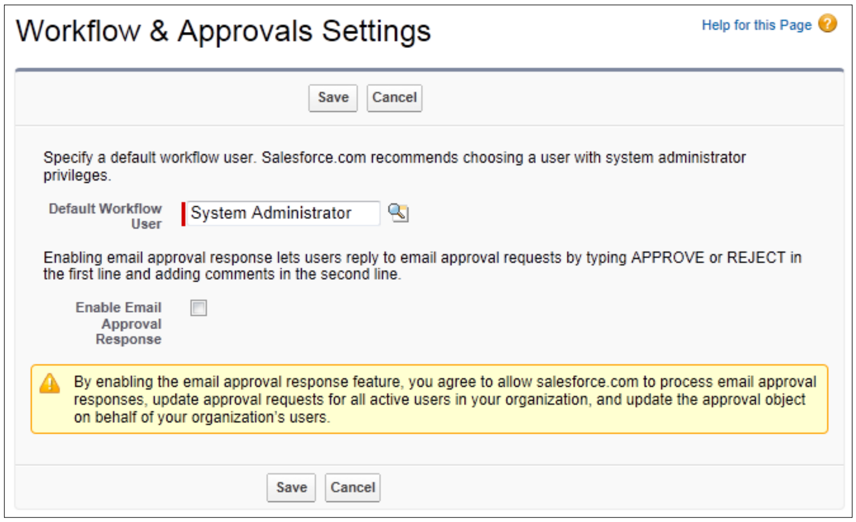


22. Finally, click on **Activate**.
23. Set **Default Workflow User** if the message that is shown in the following screenshot is shown:

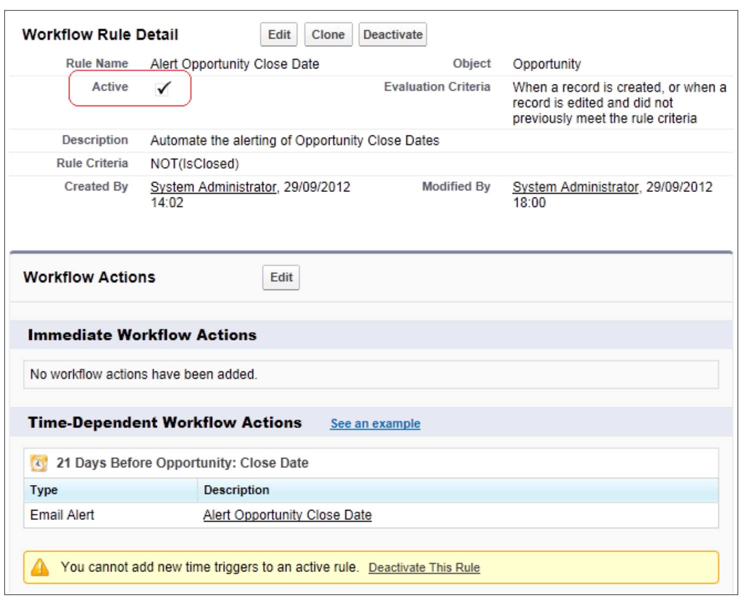


24. Click on **OK**.

We are now redirected to the **Workflow & Approvals Settings** page where we can set **Default Workflow User** (as shown in the following screenshot).



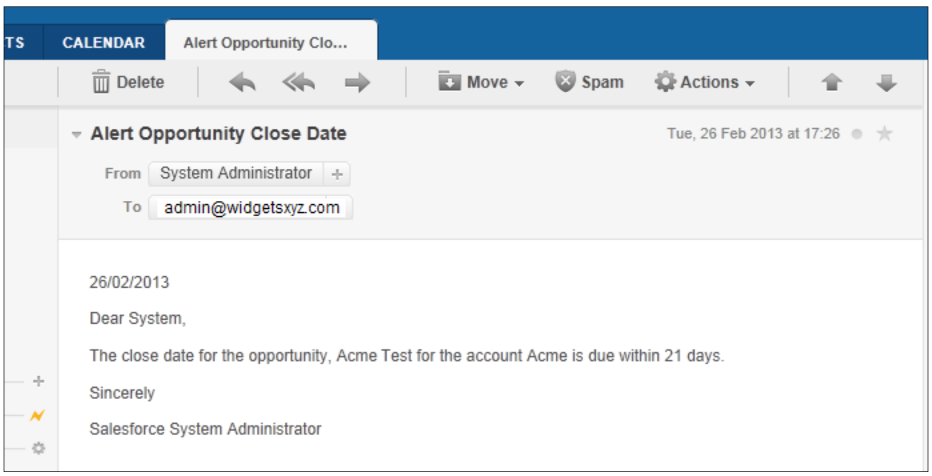
25. Set **Default Workflow User**.
26. Click on **Save**.



We are now finally redirected to the **Workflow Rule Detail** page with the **Active** checkbox checked as shown in the preceding screenshot.

How it works...

An automatic e-mail alert is now triggered using a Salesforce workflow rule that sends the e-mail alert when nearing the opportunity close date. In this example recipe, we see that an e-mail alert is received 21 days before the opportunity close date, as shown in the following screenshot:



If the opportunity close date is less than 21 days in the future, the e-mail alert will be sent immediately.

Setting a default opportunity name using a field update workflow

Often organizations use a naming convention when entering data in the CRM system.

Sales managers often instruct their sales team to enter the name of opportunities as a combination of the account name and the deal name, to help manage the processing and improve the readability of the opportunity record. For example, `Acme Opportunity X`.

This business requirement, to have an agreed format for the naming of opportunities, is a good example of where automation can be utilized in Salesforce CRM.

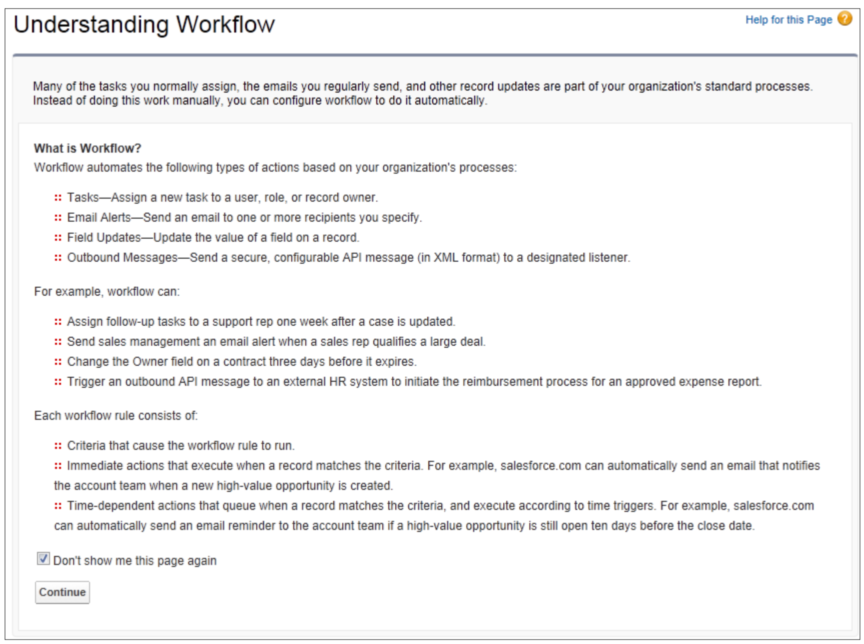
In order to automatically set the default naming of opportunities in Salesforce, a workflow rule, and a field update can be created by using this recipe.

How to do it...

Carry out the following steps to set a default opportunity name using a field update workflow:

1. Navigate to the workflow setup page, by going to **Your Name** | **Setup** | **Create** | **Workflow & Approvals**.
2. Click on **Continue** (if shown).

   The **Continue** button is found on the **Understanding Workflow** page (if this is shown) as in the following screenshot:



3. Click on **New Rule**.

> Here, we are presented with the **Step 1: Select object** page.

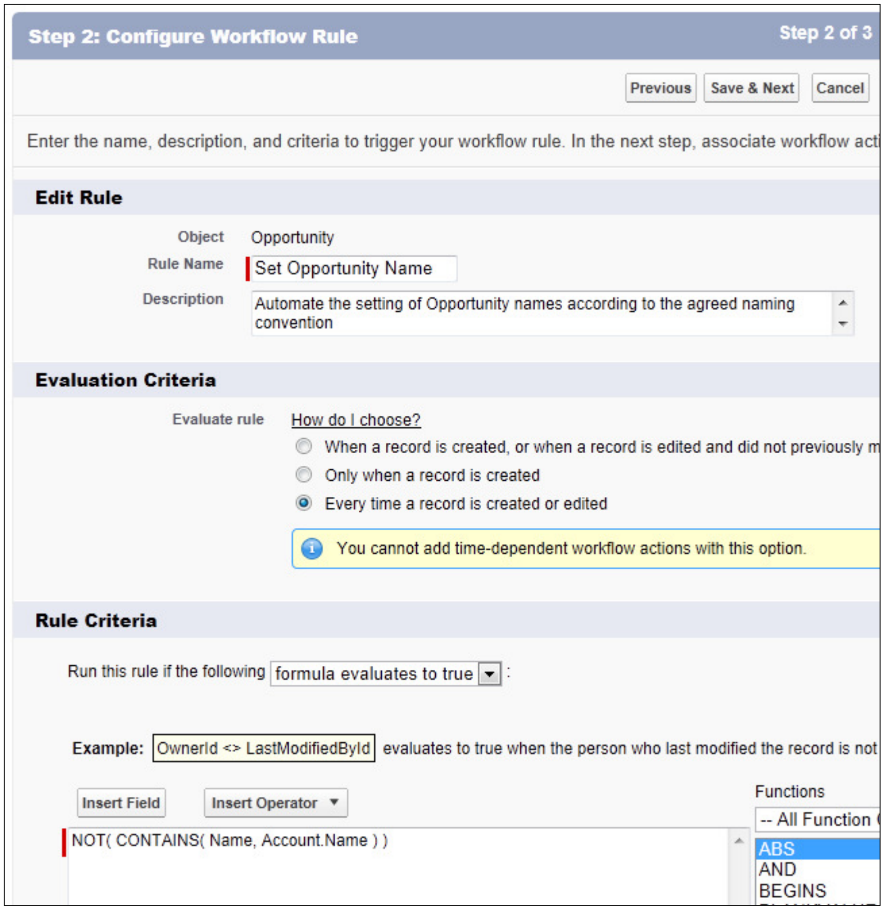
4. Select **Opportunity** and click on **Next** as shown in the following screenshot:

Here, we are presented with the **Step 2: Configure Workflow Rule** page.

5. Enter the name of the new workflow rule in the **Rule Name** field. For this recipe, type the text `Set Opportunity Name`.
6. Optionally, enter a description for the new workflow rule in the **Description** field. For this recipe, type the text `Automate the setting of Opportunity names according to the agreed naming convention`.
7. Select the **Every time a record is created or edited** option from the **Evaluation Criteria** options list.
8. In the **Rule Criteria** section, set the **Run this rule if the following** option to **formula evaluates to true**.
9. Enter the following formula in the formula editor:

```
NOT( CONTAINS( Name, Account.Name ) )
```

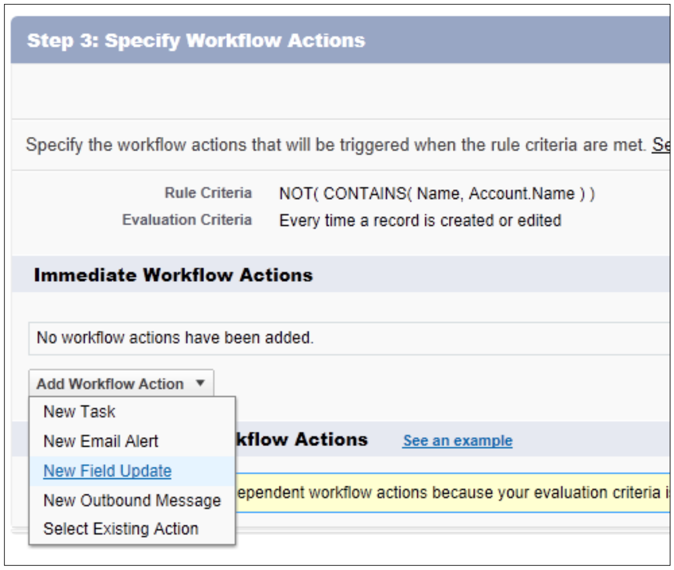
This is shown in the following screenshot:



10. Click on **Save & Next**.

> Here, we are presented with the **Step 3: Specify Workflow Actions** page.

11. In the **Immediate Workflow Actions** section, set the **Add Workflow Action** option to **New Field Update** as shown in the following screenshot:



[Here, we are presented with the **New Field Update** page.]

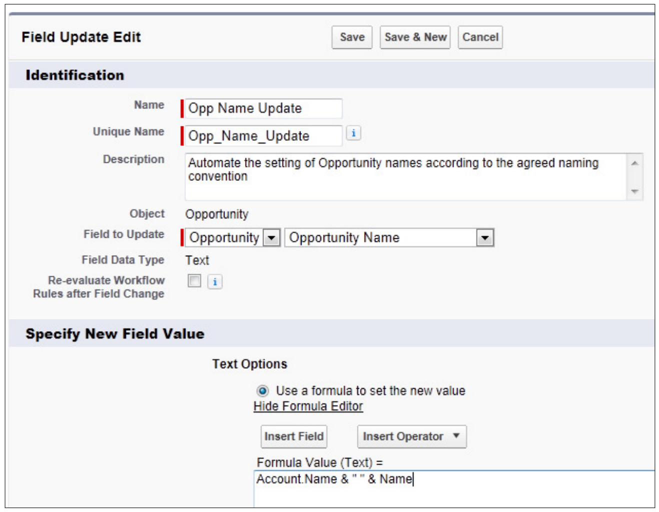
12. Enter the name for the new field update in the **Name** field. For this recipe, type the text `Opp Name Update`.
13. Accept the default value in the **Unique Name** field which is automatically set as **Opp_Name_Update**.
14. Optionally, enter a description for the new field update in the **Description** field. In this recipe, type the text `Automate the setting of Opportunity names according to the agreed naming convention`.
15. Set the **Field to Update** option to **Opportunity Name**.

[Here, we are presented with the **Specify New Field Value** section.]

16. Select the option **Use a formula to set the new value**.
17. In the formula editor box, enter the following formula:

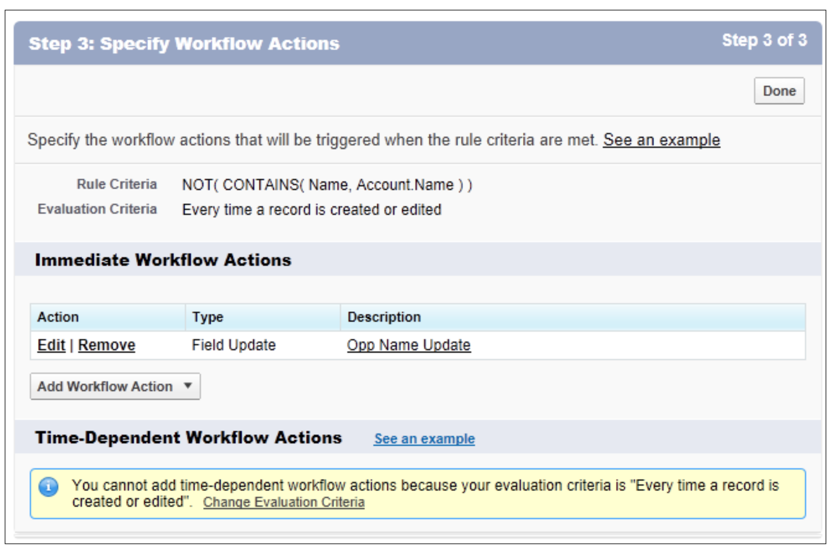
```
Account.Name & " " & Name
```

This is shown in the following screenshot:



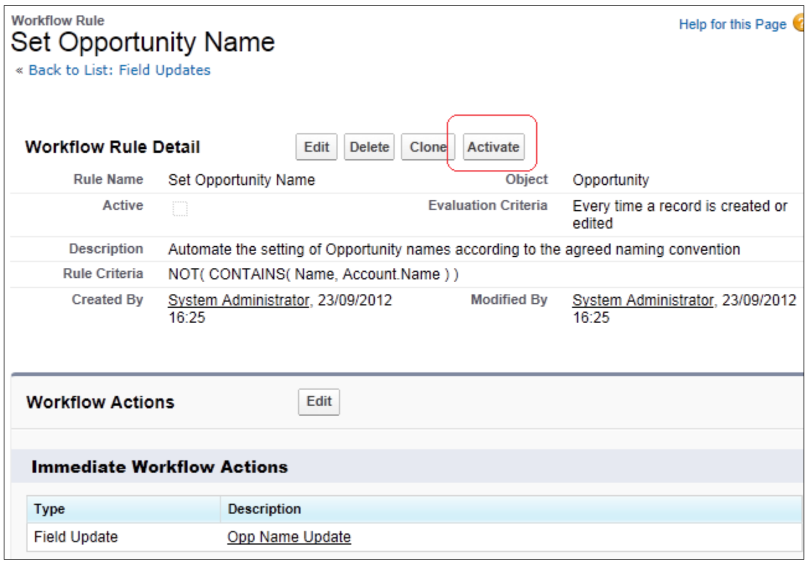
18. Click on **Save**.

> Here, we are re-shown the **Step 3: Specify Workflow Actions** page.



19. Click on **Done**.

> We are now redirected to the new workflow rule where we can activate the rule.

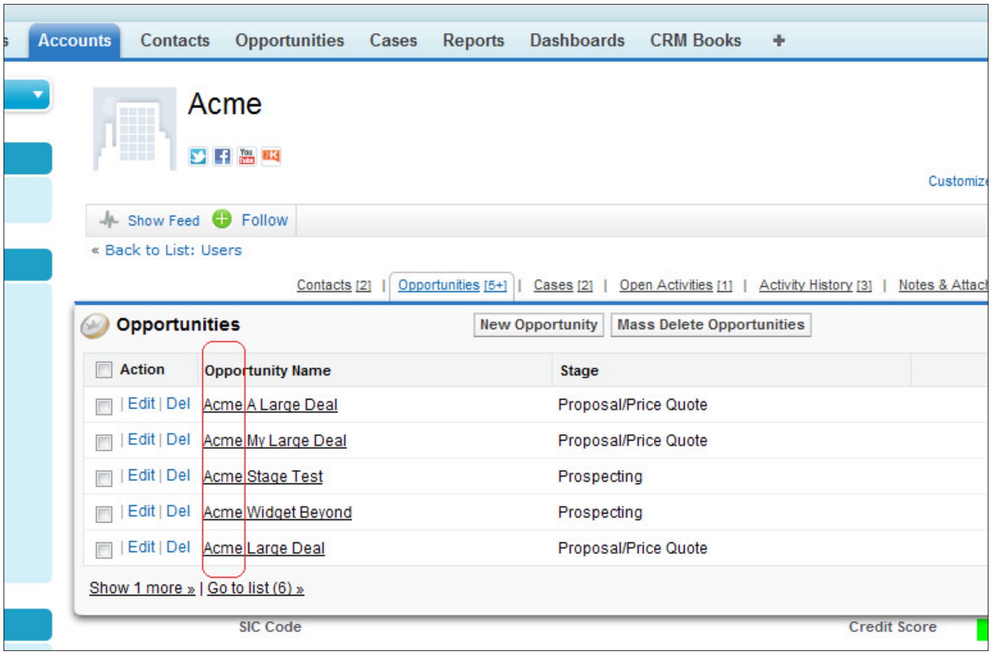


20. Finally, click on **Activate**.

How it works...

Every time an opportunity is created or edited in Salesforce CRM, the workflow rule checks if the account name appears within the text of the opportunity name.

If there is no account name text within the opportunity name, the workflow field update is immediately triggered. Here, the name of the opportunity is prefixed with the name of the account as shown in these example opportunities for the Acme account:



Generating a default opportunity close date using a field update workflow

As best practice in CRM systems, when creating opportunities there should always be a close date by which to monitor the progress of the sale.

In Salesforce, the close date is a required field but unless there is some company-specific procedure, the date could be set to anything upon creation.

Often organizations instruct their sales team to set the close date to a nominal value with respect to the creation date of the opportunity, say the date the record was created plus 21 days.

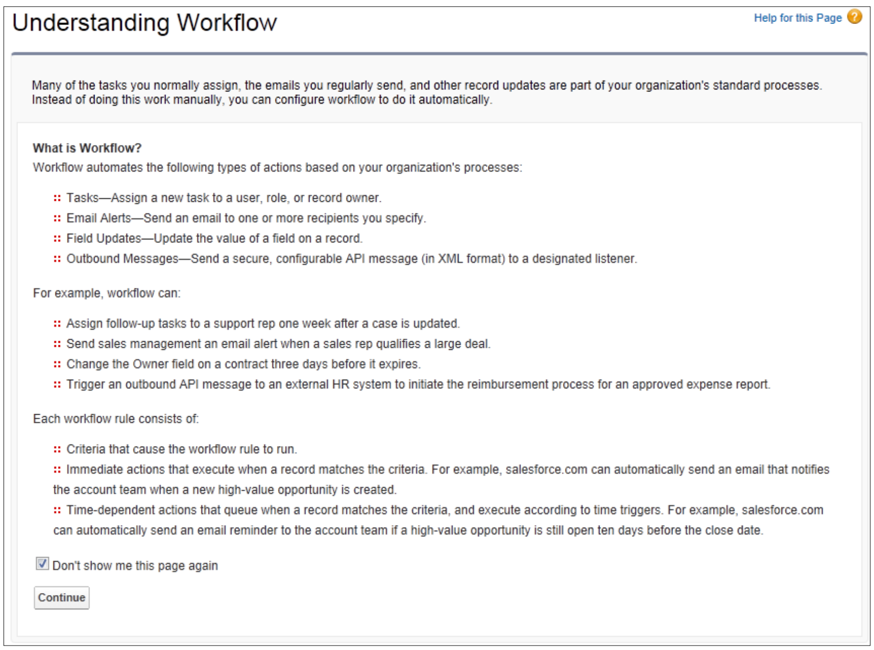
In order to set this default close date for opportunities in Salesforce, a workflow rule, a field update, and a custom new opportunity button that calls a Visualforce page, can be created using this recipe.

How to do it...

Carry out the following steps to generate a default opportunity close date using a field update workflow:

1. Navigate to the workflow setup page, by going to **Your Name** | **Setup** | **Create** | **Workflow & Approvals**.
2. Click on **Continue** (if shown).

   The **Continue** button is found on the **Understanding Workflow** page (if this is shown) as in the following screenshot:



3. Click on **New Rule**.

> Here, we are presented with the **Step 1: Select object** page.

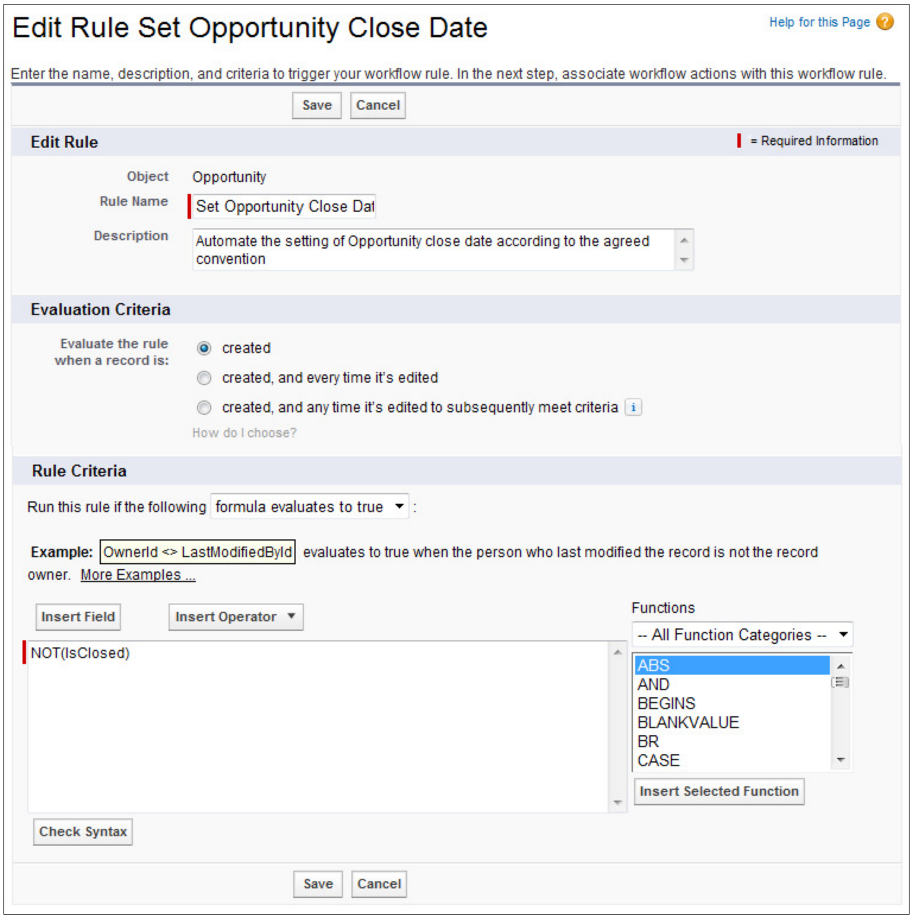
4. Select **Opportunity** and click on **Next** as shown in the following screenshot:

> Here, we are presented with the **Step 2: Configure Workflow Rule** page.

5. Enter the name of the new workflow rule in the **Rule Name** field. For this recipe, type the text `Set Opportunity Close Date`.
6. Optionally, enter a description for the new workflow rule in the **Description** field.
7. Select the **Only when a record is created** option from the **Evaluation Criteria** options list.
8. In the **Rule Criteria** section, set the **Run this rule if the following** option to **formula evaluates to true**.
9. Enter the following formula in the formula editor:

```
NOT(IsClosed)
```

This is shown in the following screenshot:



10. Click on **Save & Next**.

> Here, we are presented with the **Step 3: Specify Workflow Actions** page.

11. In the **Immediate Workflow Actions** section, set the **Add Workflow Action** option to **New Field Update**.

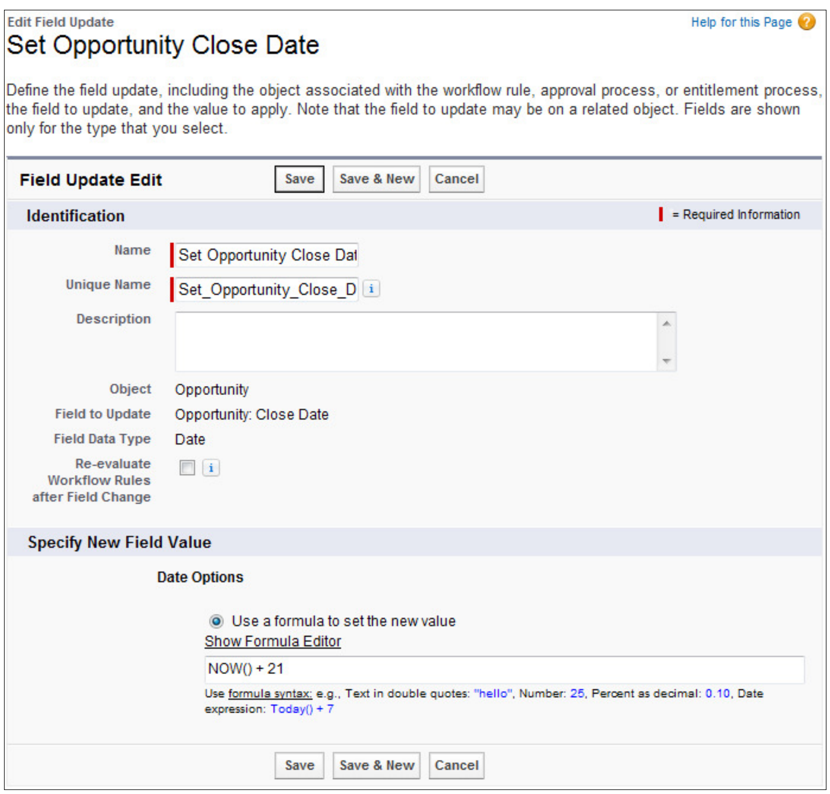
> Here, we are presented with the **New Field Update** page.

12. Enter the name for the new field update in the **Name** field. For this recipe, type the text `Set Opportunity Close Date`.
13. Accept the default value in the **Unique Name** field which is automatically set as **Set_Opportunity_Close_Date**.
14. Optionally, enter a description for the new field update in the **Description** field.
15. Set the **Field to Update** to **Opportunity CloseDate**.

> Here, we are presented with the **Specify New Field Value** section (as shown in the upcoming screenshot).

16. Select the option **Use a formula to set the new value**.
17. In the formula editor, enter the following formula (as shown in the following screenshot):

```
NOW() + 21
```



18. Click on **Save**.

> Here, we are reshown the **Step 3: Specify Workflow Actions** page.

19. Click on **Done**.

> We are now redirected to the new workflow rule where we can activate the rule.

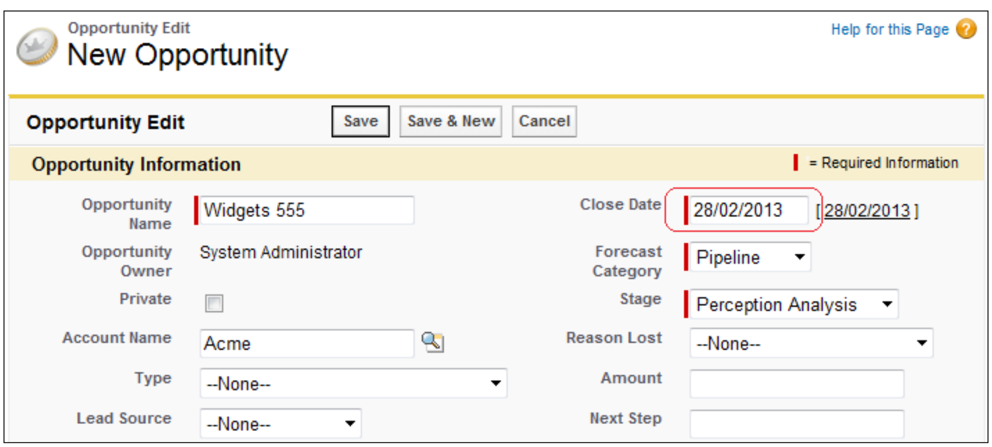
20. Finally, click on **Activate**.

How it works...

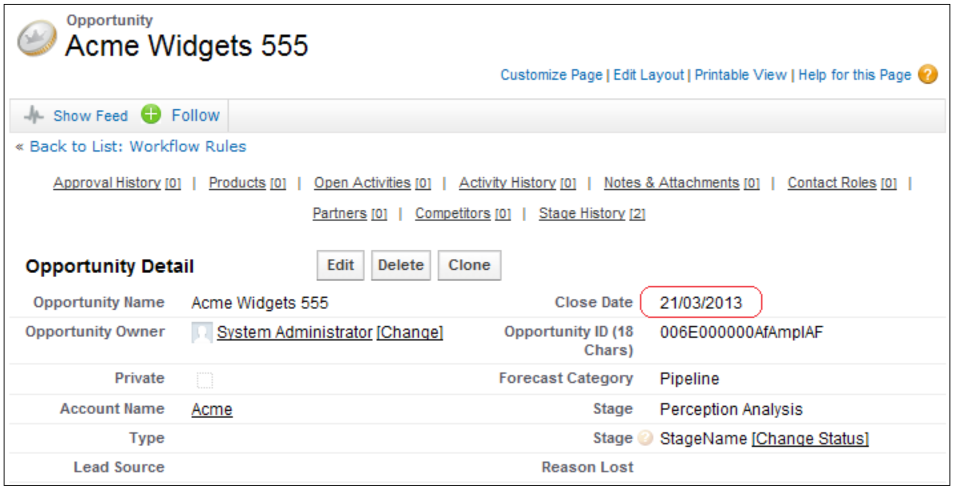
Every time an opportunity is created in Salesforce CRM, the workflow rule checks if the opportunity is not set as **Closed**.

If it is not set as **Closed**, the workflow field update is immediately triggered. Here the **Close Date** field of the opportunity is set to be now (current date and time) plus 21 days as shown in the following screenshot.

The opportunity is created and manually set with today's current date (which in this example is **28/02/2013**) as shown in the following screenshot:



Upon saving the opportunity, the workflow field update is triggered and the **Close Date** field is updated to be the current date and time plus 21 days (which in this example is **21/03/2013**) as shown in the following screenshot:



5

Improving Data Quality in Salesforce CRM

In this chapter, we will cover the following recipes:

- Stopping non-system administrators from changing account names with a validation rule
- Enforcing the use of two-letter account country codes using a validation rule
- Validating if US zip codes are well formed using a validation rule with REGEX
- Validating if UK postcodes are well formed using a validation rule with REGEX
- Creating a mandatory Reason Lost field for lost opportunities using a validation rule

Introduction

The CRM industry experts typically list poor data quality as one of the top reasons for CRM applications to fail to deliver. Poor data often leads to misleading, incomplete, and confusing information and can lead to user dissatisfaction and lack of user adoption. It is important that users understand that the data quality in a Salesforce CRM is not just the responsibility of their system administrator, but that they must also contribute to the accurate recording of information and understand how data quality directly affects their work.

It is also true that people make mistakes when entering data. This is where the power of the Salesforce CRM application and the Force.com platform becomes truly apparent. We can limit this source of error by building automatic validation and automatic workflow into the application to prevent users from making errors and cleaning up data where necessary.

Stopping non-system administrators from changing account names with a validation rule

Often there is a need to restrict the updating of data from users within Salesforce CRM. A typical example is where account records are created according to an agreed process or naming convention, which requires verification by other users or systems within the organization.

In this recipe, we will construct a validation rule that prevents the changing of account name by users who are not logged in as system administrators.

How to do it...

Carry out the following steps to create a validation rule:

1. Navigate to the account customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Validation Rules**.
2. Click on **New**.

> Here, we are presented with the **Account Validation Rule Edit** page.

3. Type `Admins Only to Update Name` in the **Field Label** textbox (upon clicking out of the textbox, the name changes to **Admins_Only_to_Update_Name**).
4. Leave the **Active** checkbox ticked.
5. Optionally, set the **Description** field to `This validation rule is for stopping non-System Administrators from changing Account Names`.

> Navigate to the **Error Condition Formula** section (as shown in the upcoming screenshot).

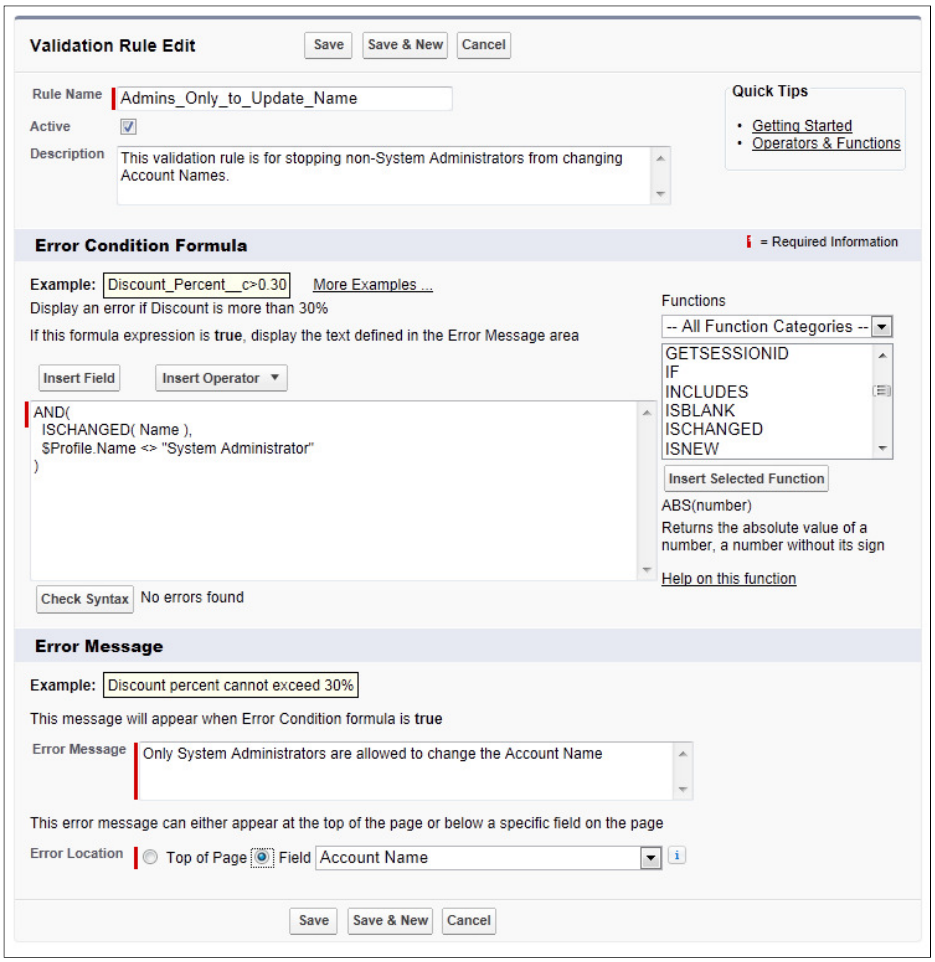
6. Paste the following code:

```
AND(
  ISCHANGED( Name ),

  $Profile.Name <> "System Administrator"
)
```

7. In the **Error Message** section, enter the text `Only System Administrators are allowed to change the Account Name`.
8. In the **Error Location** option, select **Field**.

> Here, we are presented with the **Field** selection pick list (as shown in the following screenshot).

9. Select the field **Account Name** (as shown):

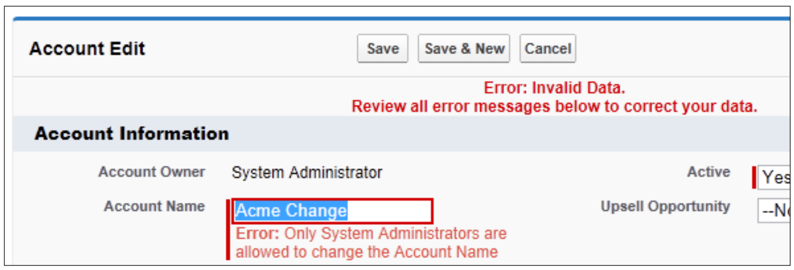


10. Finally, click on **Save**.

How it works...

Having logged in as a user that does not have a system administrator profile and changing the name of an account record, the validation rule is activated and the record is prevented from being saved. Upon the attempt to save, an error message is displayed below the **Account Name** field stating **Error: Only System Administrators are allowed to change the Account Name**.

You can see what this looks like in the following screenshot:



Enforcing the use of two-letter account country codes using a validation rule

Accurate information and reports are vital in an effective Salesforce CRM system and without it, users and sales management cannot make informed decisions. One way to achieve accurate information for reporting is by ensuring that data is stored in a consistent manner.

In this recipe, we will create a validation rule to ensure that account country codes are entered using two letters, according to an international standard (ISO 3166), from the International Organization for Standardization.

How to do it...

Carry out the following step to create a validation rule:

1. Navigate to the account customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Validation Rules**.
2. Click on **New**.

> Here, we are presented with the **Account Validation Rule Edit** page.

3. Type `Valid Billing Country` in the **Field Label** textbox (upon clicking out of the textbox, the name changes to **Valid_Billing_Country**).

4. Leave the **Active** checkbox ticked.
5. Optionally, set the **Description** field to `This validation rule is for ensuring the entry of valid billing country codes.`

> Navigate to the **Error Condition Formula** section.

6. Paste the following code:

```
OR(
ISBLANK(BillingCountry),
LEN(BillingCountry) < 2,
NOT(
CONTAINS(
"AF:AX:AL:DZ:AS:AD:AO:AI:AQ:AG:AR:AM:" &
"AW:AU:AZ:BS:BH:BD:BB:BY:BE:BZ:BJ:BM:BT:BO:" &
"BA:BW:BV:BR:IO:BN:BG:BF:BI:KH:CM:CA:CV:KY:" &
"CF:TD:CL:CN:CX:CC:CO:KM:CG:CD:CK:CR:CI:HR:" &
"CU:CY:CZ:DK:DJ:DM:DO:EC:EG:SV:GQ:ER:EE:ET:FK:" &
"FO:FJ:FI:FR:GF:PF:TF:GA:GM:GE:DE:GH:GI:GR:GL:" &
"GD:GP:GU:GT:GG:GN:GW:GY:HT:HM:VA:HN:HK:HU:" &
"IS:IN:ID:IR:IQ:IE:IM:IL:IT:JM:JP:JE:JO:KZ:KE:KI:" &
"KP:KR:KW:KG:LA:LV:LB:LS:LR:LY:LI:LT:LU:MO:MK:" &
"MG:MW:MY:MV:ML:MT:MH:MQ:MR:MU:YT:MX:FM:MD:MC:" &
"MC:MN:ME:MS:MA:MZ:MM:MA:NR:NP:NL:AN:NC:NZ:NI:" &
"NE:NG:NU:NF:MP:NO:OM:PK:PW:PS:PA:PG:PY:PE:PH:" &
"PN:PL:PT:PR:QA:RE:RO:RU:RW:SH:KN:LC:PM:VC:WS:" &
"SM:ST:SA:SN:RS:SC:SL:SG:SK:SI:SB:SO:ZA:GS:ES:" &
"LK:SD:SR:SJ:SZ:SE:CH:SY:TW:TJ:TZ:TH:TL:TG:TK:" &
"TO:TT:TN:TR:TM:TC:TV:UG:UA:AE:GB:US:UM:UY:UZ:" &
"VU:VE:VN:VG:VI:WF:EH:YE:ZM:ZW",
BillingCountry)))
```

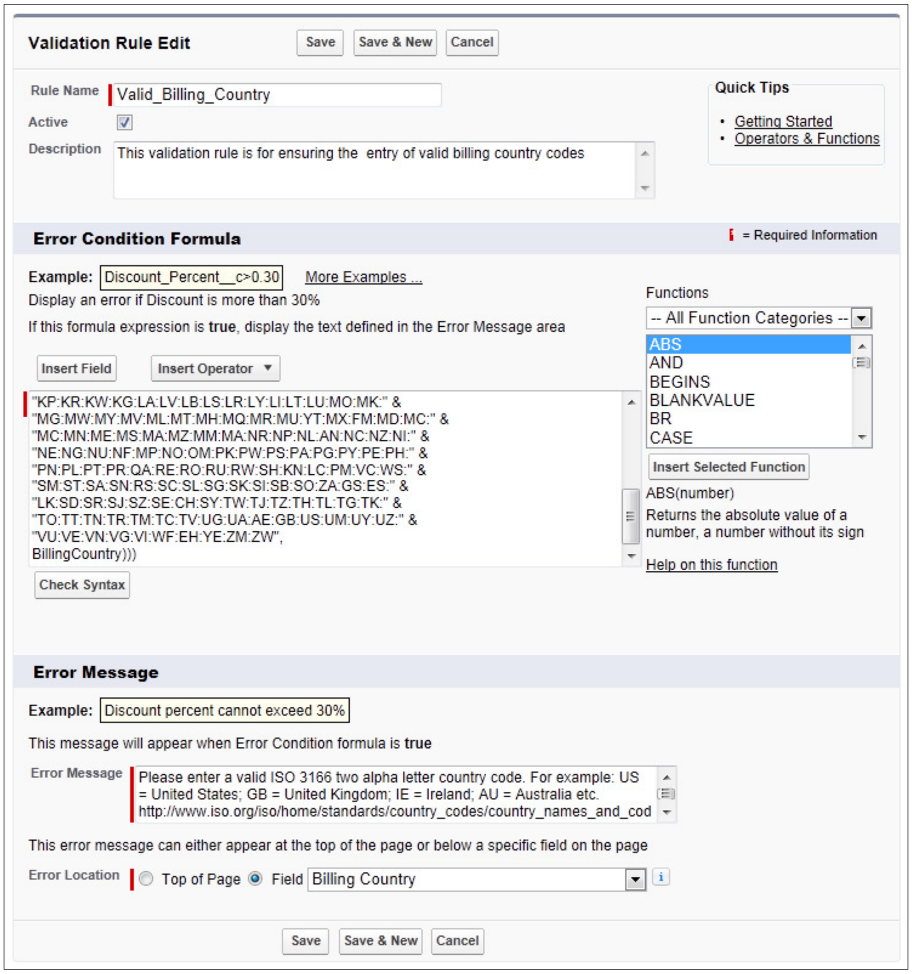
7. In the **Error Message** section, enter the following text:

```
Please enter a valid ISO 3166 two alpha letter country code. For
example: US = United States; GB = United Kingdom; IE = Ireland; AU
= Australia etc.
http://www.iso.org/iso/home/standards/country_codes/country_
names_and_code_elements.htm
```

8. In the **Error Location** option, select **Field**.

> Here, we are presented with the **Field** selection pick list.

9. Select the field **Billing Country** (as shown):



10. Finally, click on **Save**.

How it works...

You can see how the entering of an invalid country code is validated and how the message appears within the edit page as in the following screenshot:

Validating if US zip codes are well formed using a validation rule with REGEX

Accurate information and reports are vital in an effective Salesforce CRM system and without it, users and sales management cannot make informed decisions. One way to achieve accurate information for reporting is by ensuring that data is stored in a consistent manner.

In this recipe, we will create a validation rule to ensure that US zip codes conform to the correct format.

How to do it...

Carry out the following step to create a validation rule:

1. Navigate to the account customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Validation Rules**.
2. Click on **New**.

> Here, we are presented with the **Account Validation Rule Edit** page.

3. Type `Valid US Zip Code` in the **Field Label** textbox (upon clicking out of the textbox, the name changes to **Valid_US_Zip_Code**).
4. Leave the **Active** checkbox checked.
5. Optionally, set the **Description** field to `This validation rule is for ensuring the entry of valid US Zip codes.`

> Navigate to the **Error Condition Formula** section.

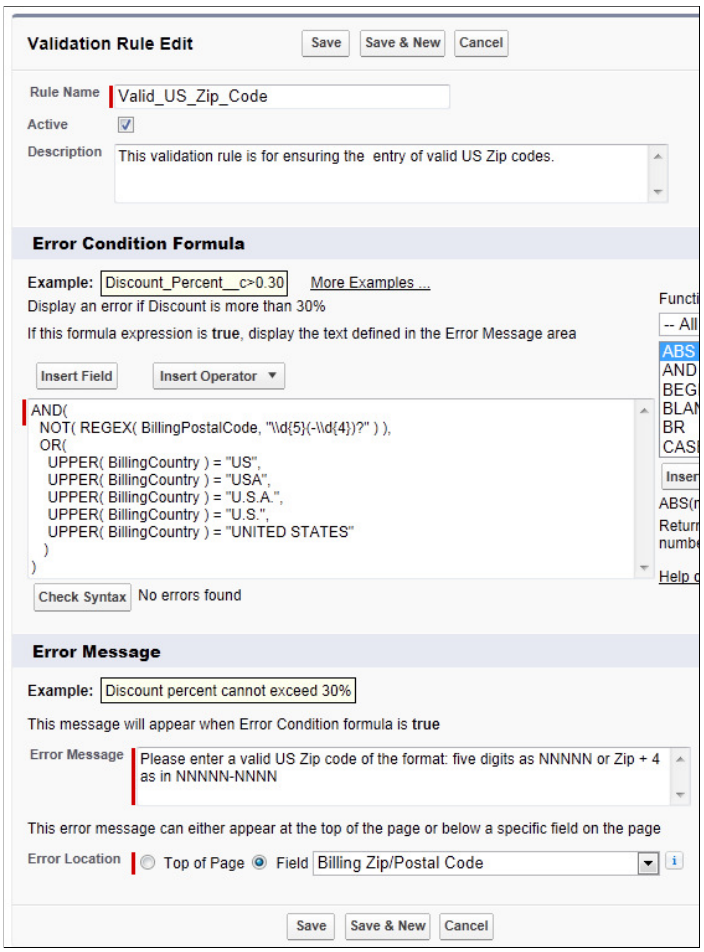
6. Paste the following code:

```
AND(
  NOT( REGEX( BillingPostalCode, "\\d{5}(-\\d{4})?" ) ),
  OR(
    UPPER( BillingCountry ) = "US",
    UPPER( BillingCountry ) = "USA",
    UPPER( BillingCountry ) = "U.S.A.",
    UPPER( BillingCountry ) = "U.S.",
    UPPER( BillingCountry ) = "UNITED STATES"
   )
)
```

7. In the **Error Message** section, enter the text `Please enter a valid US Zip code of the format: five digits as NNNNN or Zip + 4 as in NNNNN-NNNN`.
8. In the **Error Location** option, select **Field**.

> Here, we are presented with the **Field** selection pick list.

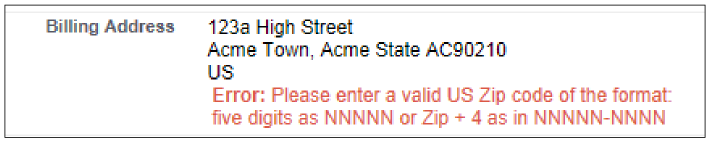
9. Select the field **Billing Zip/PostalCode** (as shown):



10. Finally, click on **Save**.

How it works...

You can see how the entering of an invalid US zip code is validated and how the message appears within the edit page in the following screenshot:



Validating if UK postcodes are well formed using a validation rule with REGEX

Accurate information and reports are vital in an effective Salesforce CRM system and without it, users and sales management cannot make informed decisions. One way to achieve accurate information for reporting is by ensuring that data is stored in a consistent manner.

In this recipe, we will create a validation rule to ensure that UK postcodes conform to the correct format.

How to do it...

Carry out the following step to create a validation rule:

1. Navigate to the **Accounts** customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Validation Rules**.
2. Click on **New**.

> Here, we are presented with the **Account Validation Rule Edit** page.

3. Type `Valid UK Postcode` in the **Field Label** textbox (upon clicking out of the text box, the name changes to **Valid_UK_Postcode**).
4. Leave the **Active** checkbox checked.
5. Optionally, set the **Description** field to `This validation rule is for ensuring the entry of valid UK Postcodes.`

> Navigate to the **Error Condition Formula** section.

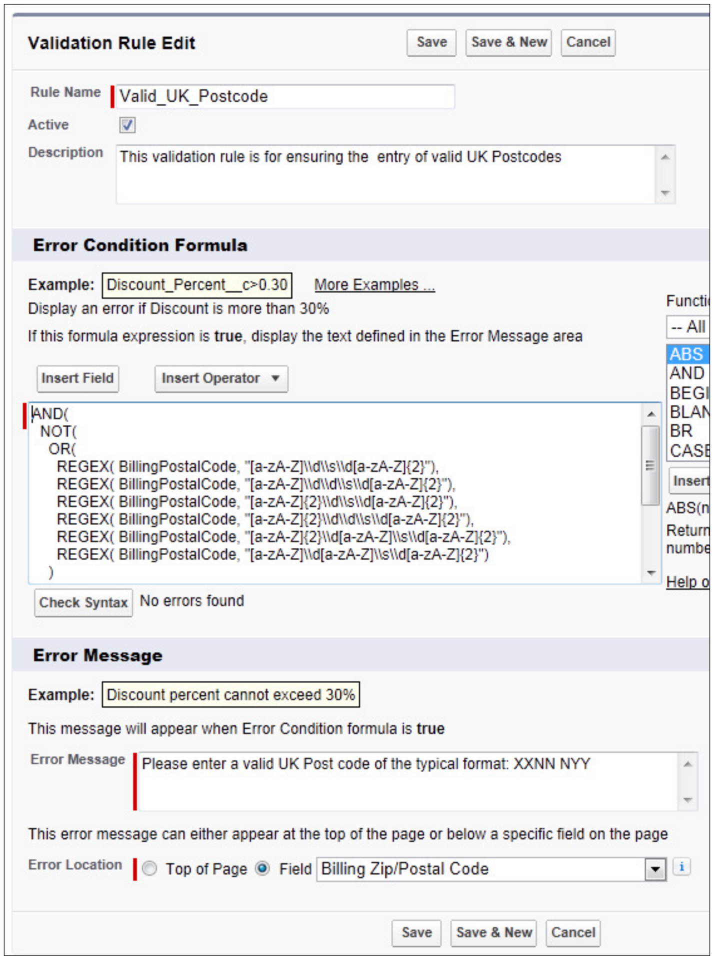
6. Paste the following code:

```
AND(
  NOT(
    OR(
      REGEX( BillingPostalCode, "[a-zA-Z]\\d\\s\\d[a-zA-Z]{2}"),
      REGEX( BillingPostalCode, "[a-zA-Z]\\d\\d\\s\\d[a-zA-Z]
{2}"),
      REGEX( BillingPostalCode, "[a-zA-Z]{2}\\d\\s\\d[a-zA-Z]
{2}"),
      REGEX( BillingPostalCode, "[a-zA-Z]{2}\\d\\d\\s\\d[a-zA-Z]
{2}"),
      REGEX( BillingPostalCode, "[a-zA-Z]{2}\\d[a-zA-Z]\\s\\d[a-
zA-Z]{2}"),
      REGEX( BillingPostalCode, "[a-zA-Z]\\d[a-zA-Z]\\s\\d[a-zA-Z]
{2}")
    )
  ),
  OR(
    UPPER( BillingCountry ) = "UK",
    UPPER( BillingCountry ) = "U.K.",
    UPPER( BillingCountry ) = "UNITED KINGDOM"
  )
)
```

7. In the **Error Message** section, enter the text `Please enter a valid UK Post code of the typical format: XXNN NYY`.
8. In the **Error Location** option, select **Field**.

> Here, we are presented with the **Field** selection pick list.

9. Select the field **Billing Zip/PostalCode** (as shown):



10. Finally, click on **Save**.

How it works...

You can see how the entering of an invalid UK postcode is validated and how the message appears within the edit page in the following screenshot:

Creating a mandatory Reason Lost field for lost opportunities using a validation rule

In this recipe, we will create a conditional validation rule that requires the entry of the reason of why an opportunity has been lost.

The reason lost is a custom field (which we created as `Reason_Lost__c`) and is first created using the steps of this recipe.

Getting ready

Creating a custom Reason Lost field on the opportunity:

1. Navigate to the Opportunity Fields setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Fields**.

   > Scroll down to the **Opportunities Custom Fields & Relationships** section.

2. Click on **New**.

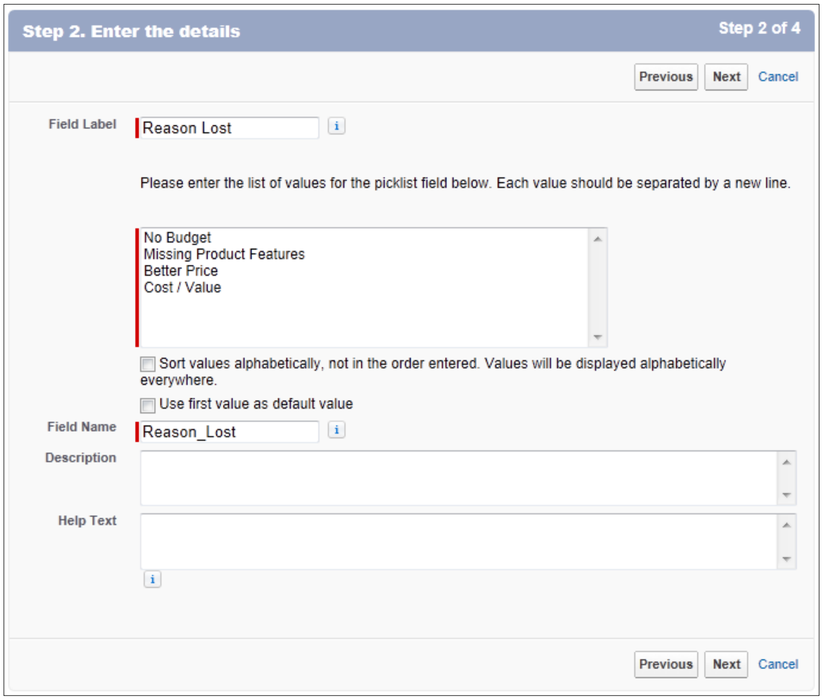
   > You will find the **Step 1. Choose the field type** page.

3. Choose **Picklist** from the **Data Type** options.
4. Click on **Next**.

   > The **Step 2. Enter the details** page is presented.

5. Type `Reason Lost` in the **Field Label** textbox (as shown in the screenshot).
6. Accept the default **Reason_Lost** in the **Name** field.
7. Type these example values into the pick list field:

   ```
   No Budget
   Missing Product Features
   Better Price
   Cost / Value
   ```



8. Click on **Next**.

> Next, you will see the **Step 3. Establish field-level security** page.

9. Click on **Next**.
10. Select the profiles to which you want to grant edit or view access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
11. Click on **Save**.

> Next, you will see the **Step 4. Add to page layouts** page.

12. Click on **Save**.

To create the composite validation rule, we need to ensure that the custom field (**Reason_Lost__c**) has been completed whenever the opportunity has been set to **Closed Lost** by following these steps:

How to do it...

Carry out the following step to create a validation rule:

1. Navigate to the opportunity customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Validation Rules**.
2. Click on **New**.

> Here, we are presented with the **Opportunity Validation Rules** edit page.

3. Type `Opportunity Reason Lost` in the **Field Label** textbox (upon clicking out of the text box, the name changes to **Opportunity_Reason_Lost**).
4. Leave the **Active** checkbox ticked.
5. Optionally, set the **Description** field to `This validation rule is for ensuring Reason Lost is completed when Opportunity is Lost`.

> Navigate to the **Error Condition Formula** section (as shown in the upcoming screenshot).

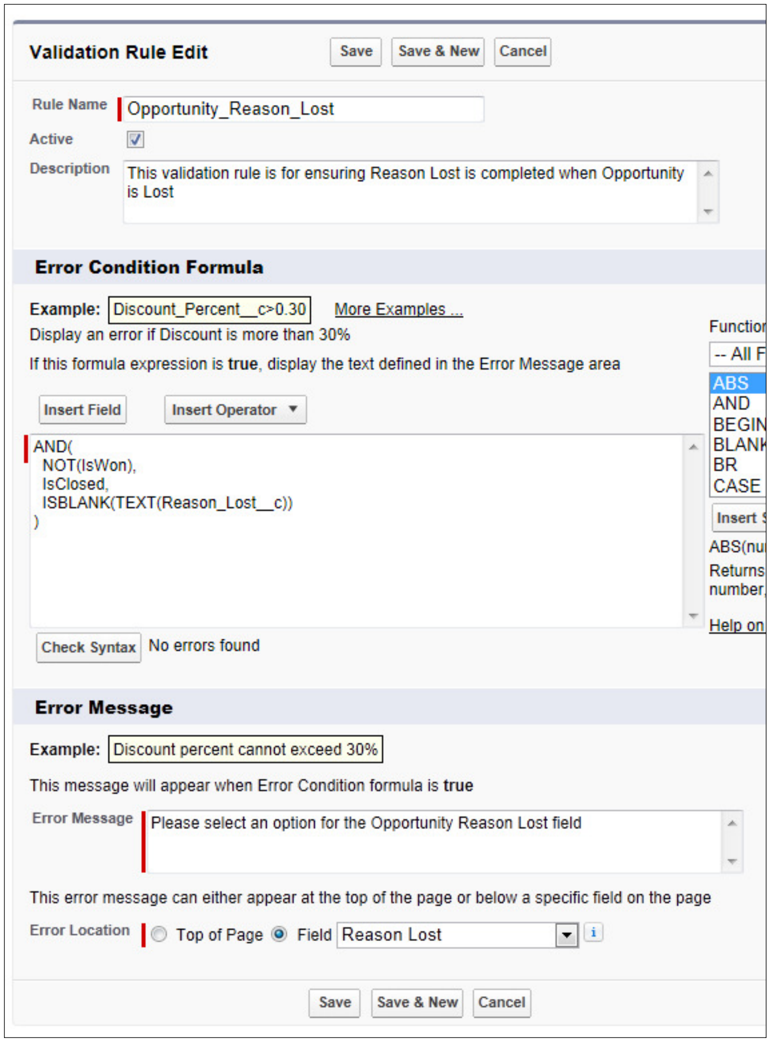
6. Paste the following code:

```
AND(
  NOT(IsWon),
  IsClosed,
  ISBLANK(TEXT(Reason_Lost__c))
)
```

7. In the **Error Message** section, enter the text `Please select an option for the Opportunity Reason Lost field`.
8. In the **Error Location** option, select **Field**.

> Here, we are presented with the **Field** selection pick list.

9. Select the field **Reason Lost** (as shown):

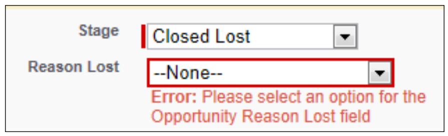


10. Finally, click on **Save**.

How it works...

You can see how the setting of the opportunity sales stage changes to **Closed Lost** and the attempted save then fires the validation rule.

Here, the user must enter the reason that the opportunity was lost in the custom field created in this recipe called **Reason Lost** as shown in the following screenshot:



6

Implementing Approval Processes

In this chapter, we will cover the following recipes:

- Creating an e-mail template for use with approval assignments
- Setting up a user to be associated with an approval process
- Building an opportunity approval process for deals greater than USD 100,000

Introduction

An approval process in Salesforce CRM is an automated mechanism that you can set up to automate and control the approval of records within your organization.

Approval processes contain a structured set of steps that are used to facilitate the review and action of records that match a specified criterion where approval is required. The approval process also allows for the specifying of which users are to approve the record at each step.

Each step of an approval process can apply to either all records within the process or specified records that have certain field values.

The building of approval processes involves the setting of what actions are to be taken after the record is either first submitted, approved, rejected, or recalled from the approval process. The setup of an approval process also allows for the selective locking of the record during the process.

Approval processes provide you and your organization with a powerful mechanism to control internal processes that must be completed as part of an organization business process requirement. For example, setting up line manager approval for leave of absence, ensuring that financial managers agree with budgets and planned expenditure, sanctioning marketing campaigns, or authorizing that large sales deals are acceptable to proceed.

Creating an e-mail template for use with approval assignments

When building approval processes for records in Salesforce CRM, it is necessary to specify the users that are to approve the record at each step of the approval process.

To notify these users that an approval request has been assigned to them, an e-mail is sent by the system. Here, you can specify a custom e-mail template to be used when notifying an approver.

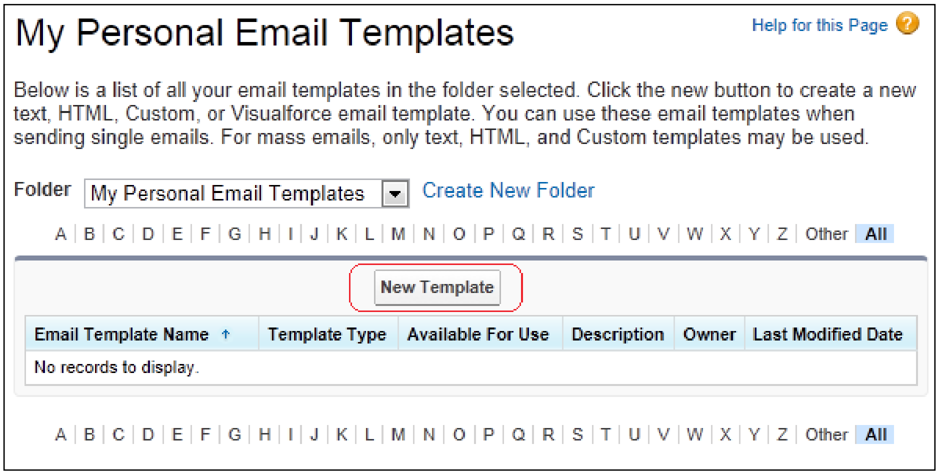
In this recipe, we will create an e-mail template that will be used to notify users that an opportunity is at or above a certain amount and at a specified stage and that the record requires their approval to continue through the sales process.

How to do it...

Carry out the following steps to create an e-mail template for use with approval assignments:

1. Navigate to the new e-mail template creation page, by going to **Your Name** | **Setup** | **Email** | **My Templates**.
2. Select the folder **My Personal Email Templates**.

3. Click on the **New Template** button as shown in the following screenshot:



[Here, we are presented with the **Step 1. Email Template: New Template** page.]

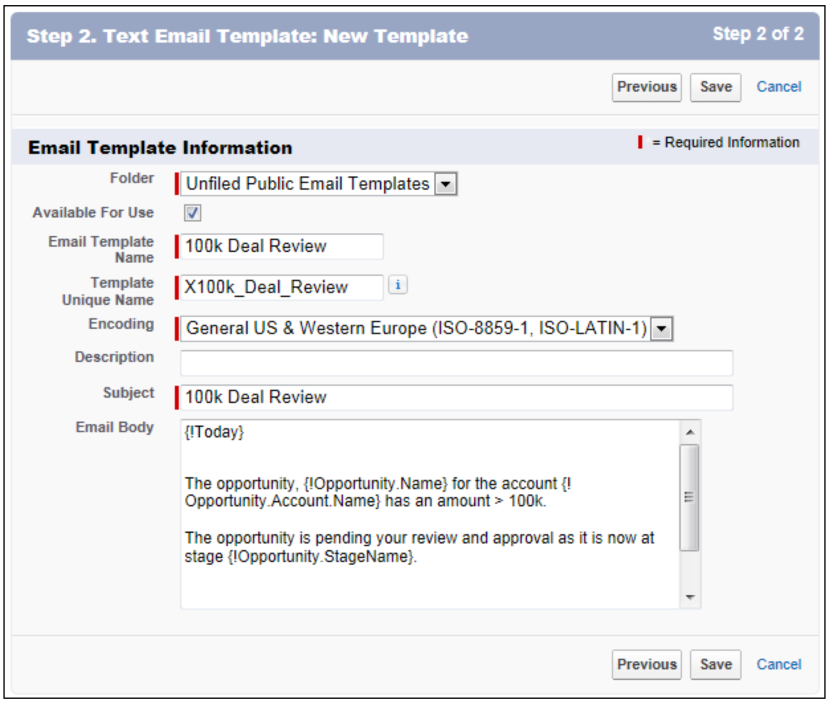
4. Choose the **Text** option and click on the **Next** button.

[Here, we are presented with the **Step 2. Text Email Template: New Template** page.]

5. Choose the **Folder** option **Unfiled Public Email Templates** (as shown in the upcoming screenshot).
6. Select the checkbox **Available For Use**.
7. Enter the name of the e-mail template in the **Email Template Name** field. For this recipe, type the text `100k Deal Review`.
8. Accept the default text **X100k_Deal_Review** that is automatically set in the **Template Unique Name** field.
9. Optionally, enter a description in the **Description** field.
10. Enter the subject for the e-mail in the **Subject** field. For this recipe, type the text **100k Deal Review**.

11. In the **Email Body** field, enter the following code:

```
{!Today}


The opportunity, {!Opportunity.Name} for the account
{!Opportunity.Account.Name} has an amount > 100k.

The opportunity is pending your review and approval as it is now
at stage {!Opportunity.StageName}.
```



12. Click on **Save**.

Setting up a user to be associated with an approval process

When building approval processes for records in Salesforce CRM, it is necessary to specify the users who have to approve the record at each step of the approval process. We can also restrict the approval process for specific users by setting an appropriate value on their user record.

To notify these users that an approval request has been assigned to them, an e-mail is sent by the system. Here, you can specify that a record owner's manager is to be set as the approver for the record. To restrict the approval process for specific users, you can specify that, say, only the sales team is to submit opportunity reviews.

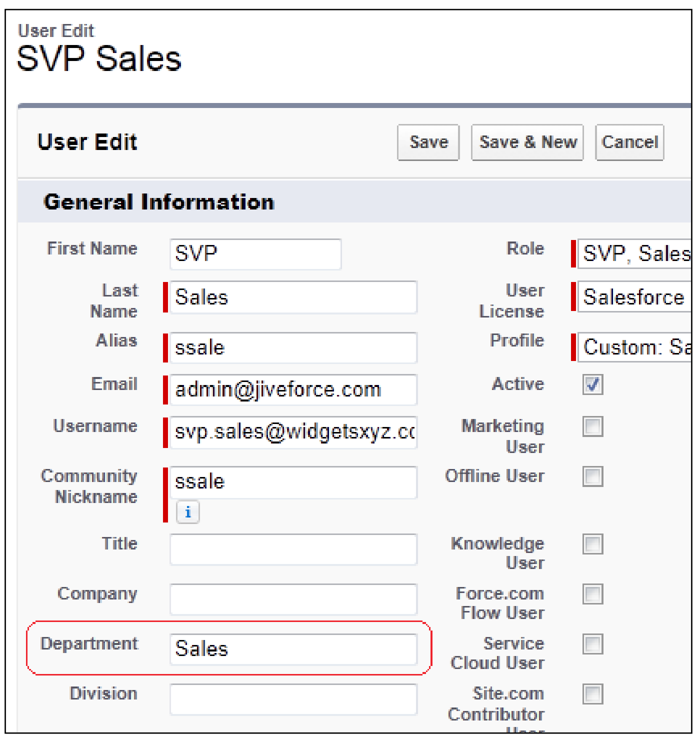
In this recipe, we will edit a user record to set the value of the **Manager** field to a user record to which approval notifications will be sent. We will also set the value of the **Department** field to **Sales** which is used as one of the criterion for the triggering of the approval process.

The sales user setting is entered in the standard **Department** field and the manager is set using the standard **Manager** lookup field and these are set using this recipe.

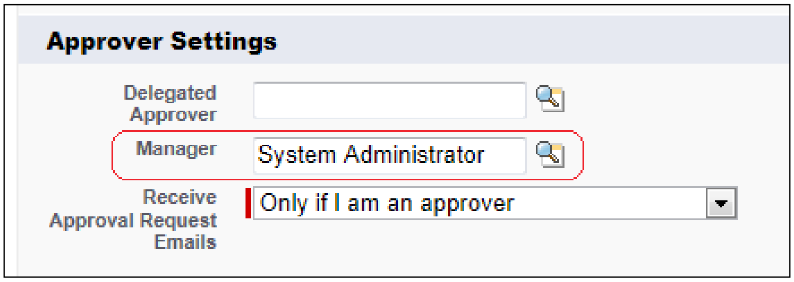
How to do it...

Carry out the following steps to set up a user that is to be associated with approval processes:

1. Navigate to the user's record page, by going to **Your Name** | **Setup** | **Manage Users** | **Users**.
2. Edit the user's record. In this recipe, we are setting the **Department** field for the user **SVP Sales** to **Sales** as shown in the following screenshot:



3. Scroll down the user's edit page to the **Approver Settings** section and select a manager for the user. In this recipe we are setting the **Manager** field for the user **SVP Sales** to **System Administrator** as shown in the following screenshot:



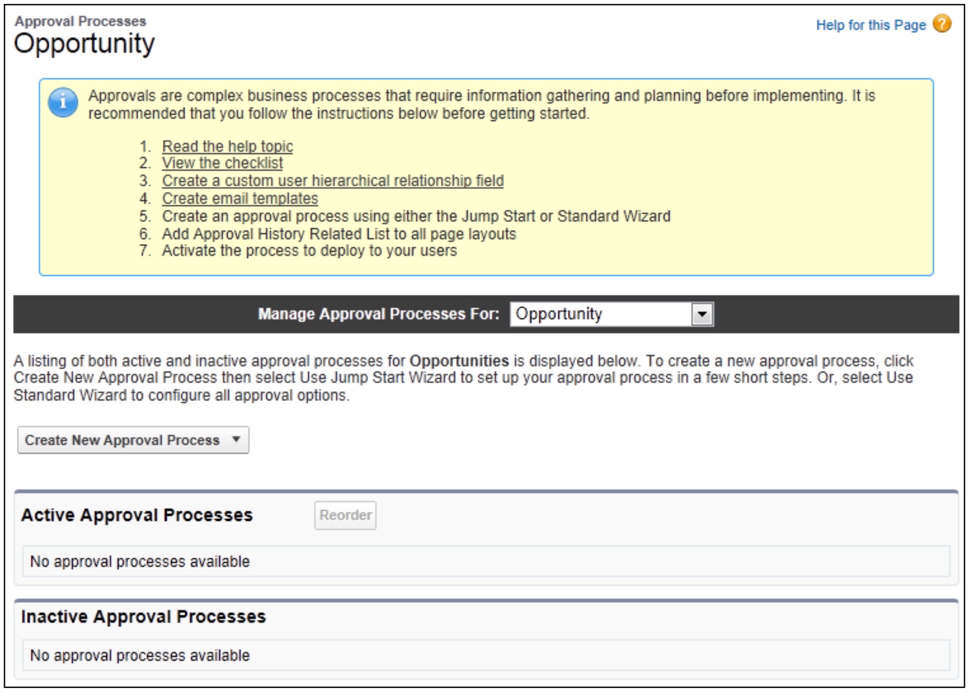
Building an opportunity approval process for deals greater than USD 100,000

In this recipe, we will build an Opportunity approval process for deals greater than USD 100,000.

How to do it...

Carry out the following steps to build an opportunity approval process for deals greater than USD 100,000:

1. Navigate to the approval processes setup page, by going to **Your Name** | **Setup** | **Create** | **Workflow & Approvals** | **Approval Processes**.
2. In the **Manage Approval Processes For:** pick list choose, **Opportunity** as shown in the following screenshot:

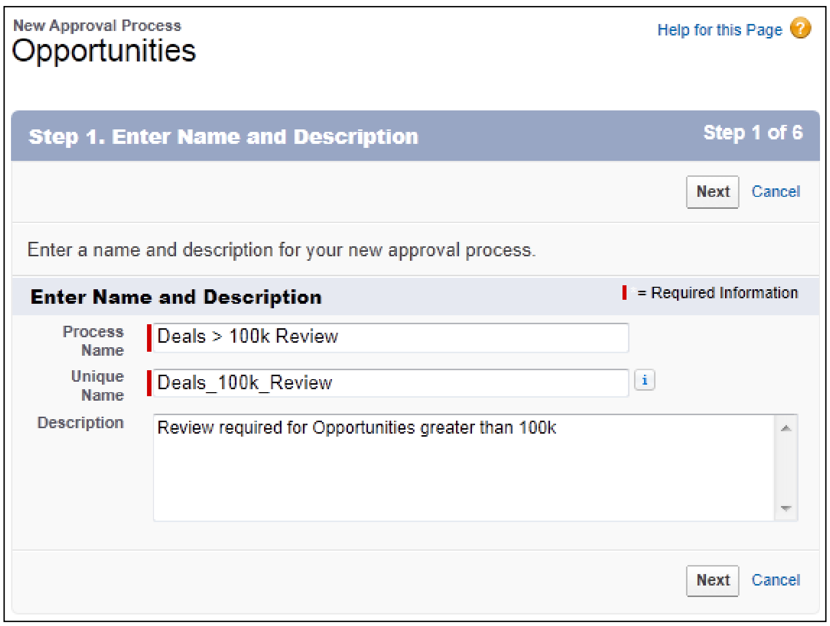


Here, we are presented with a listing of both active and inactive approval processes for opportunities as shown in the previous screenshot.

3. Click on the **Create New Approval Process** button.
4. Choose the pick list option **Use Standard Setup Wizard** (as shown in the following screenshot):

> Here, we are presented with the **Step 1. Enter Name and Description** page.

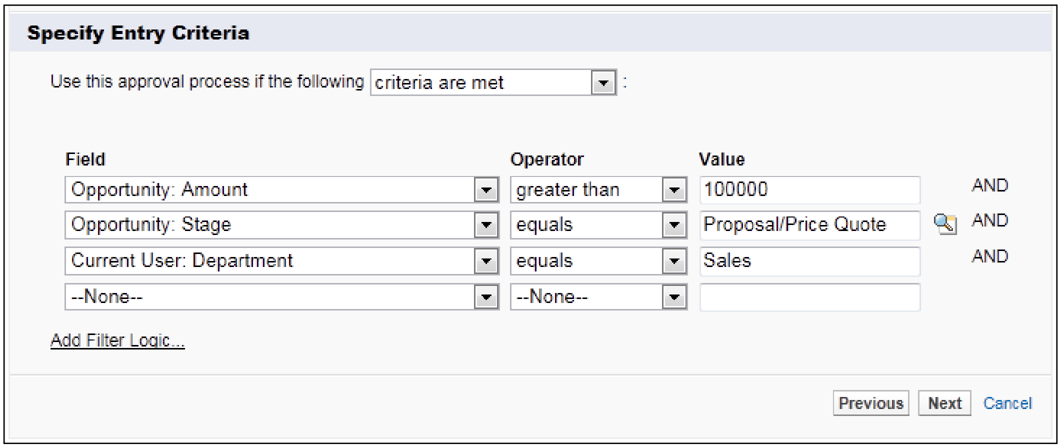
5. Enter the name of the new approval process in the **Process Name** field. For this recipe, type the text `Deals > 100k Review`.
6. The **Unique Name** field is automatically set to **Deals_100k_Review**.
7. Optionally, enter a description for the new approval process in the **Description** field. For this recipe, type the text `Review required for Opportunities greater than 100k`:



8. Click on **Next**.

> Here, we are presented with the **Step 2. Specify Entry Criteria** page.

9. Select the following field, operator, and value: **Opportunity: Amount**; **greater than**; **100000**.
10. For the second row, select the following field, operator, and value: **Opportunity: Stage**; **equals**; **Proposal/Price Quote**.
11. For the third row, select the following field, operator, and value: **Current User: Department**; **equals**; **Sales** (as shown in the following screenshot:

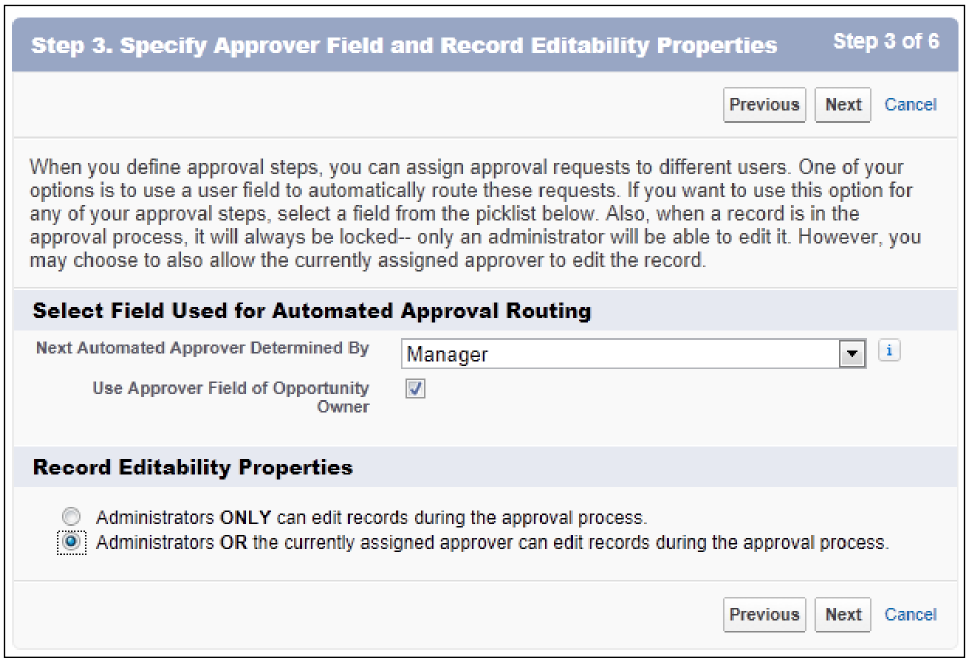


12. Click on **Next**.

> Here, we are presented with the **Step 3: Specify Approver Field and Record Editability Properties** page.

13. In the **Next Automated Approver Determined By** pick list, select the **Manager** option.
14. Set the checkbox to true for the **Use Approver Field of Opportunity Owner**.
15. In the **Record Editability Properties** section, choose the option **Administrators OR the currently assigned approver can edit records during the approval process** as shown in the following screenshot.

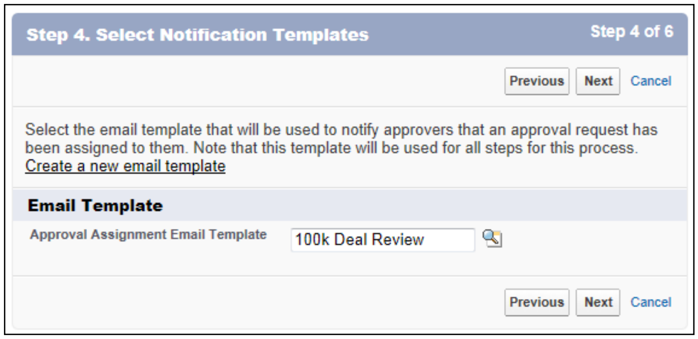
[By setting **Record Editability Properties** as shown, you are allowing the approving user to make changes to the record while the record is flowing through the approval process. This is useful as the record will remain locked to every user (with the exception of the administrators and approvers) and will prevent the need for edits to be made only by system administrators as this creates additional effort for administrators and risks delays during the sales process.]



16. Click on **Next**.

[Here, we are presented with the **Step 4. Select Notification Templates** page.]

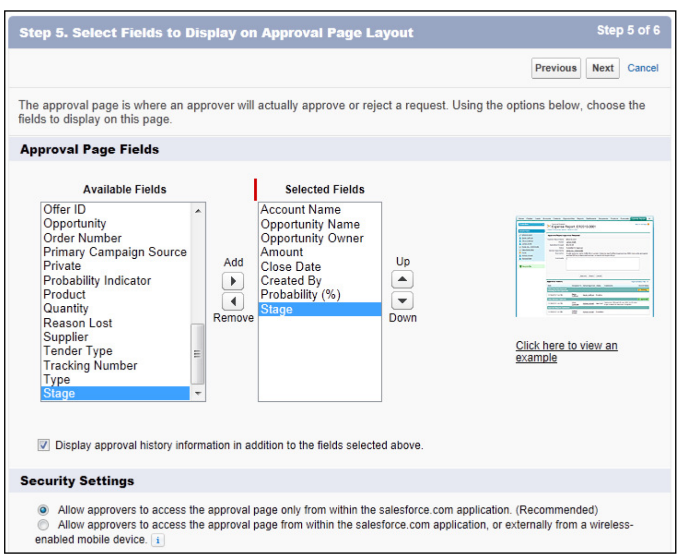
17. Select the **100k Deal Review** template in the **Approval Assignment Email Template** lookup that was created in the *Creating an e-mail template for use with approval assignments* recipe:



18. Click on **Next**.

[Here, we are presented with the **Step 5. Select Fields to Display on Approval Page Layout** page.]

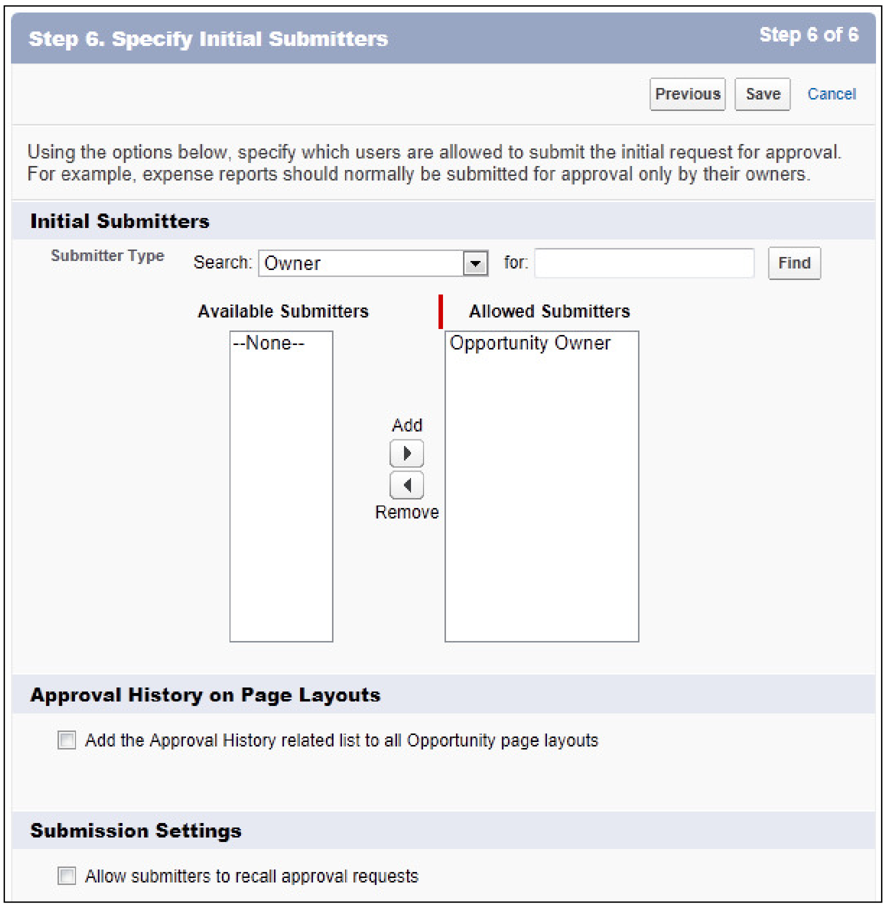
19. Select the fields to display on the approval page. Here, we are selecting to show **Account Name**, **Opportunity Name**, **Opportunity Owner**, **Amount**, **Close Date**, **Created By**, **Probability**, and **Stage**.
20. Select the checkbox **Display approval history information in addition to the fields selected above** as shown in the following screenshot:



21. Click on **Next**.

> Here, we are presented with the **Step 6. Specify Initial Submitters** page.

22. Here we are leaving the default setting of initial submitter to be set to the **Opportunity Owner** value as shown in the following screenshot:

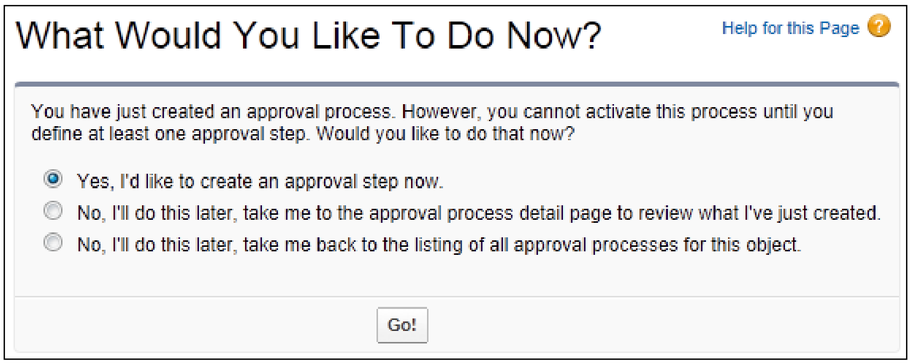


23. Click on **Save**.

Here, we are presented with the **What Would You Like To Do Now?** page.

This is required because you cannot activate an approval process until you have defined at least one approval step.

24. Select the **Yes, I'd like to create an approval step now** option as shown in the following screenshot:

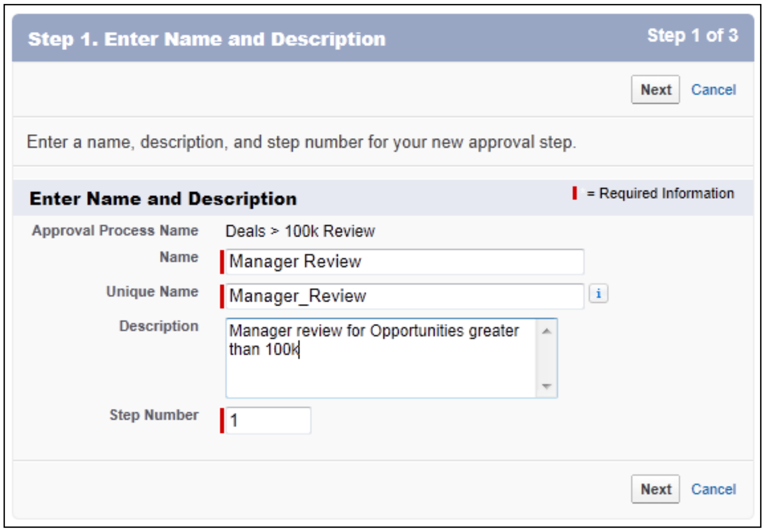


25. Click on **Go!**.

Here, we are shown the **New Approval Step** section and presented with the **Step 1. Enter Name and Description** page.

26. Enter the name of the new approval step in the **Name** field. For this recipe, type the text `Manager Review`.
27. The **Unique Name** field is automatically set to **Manager_Review**.
28. Optionally, enter a description for the new approval step in the **Description** field. For this recipe, type the text `Manager review for Opportunities greater than 100k`.

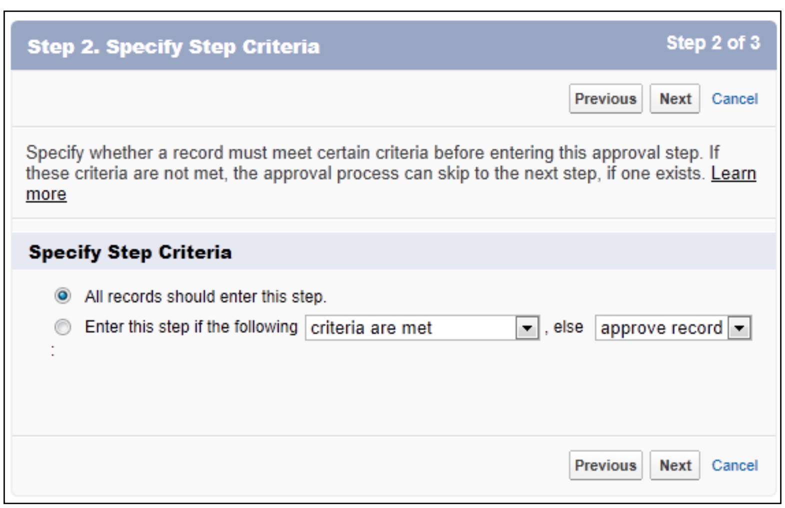
29. Leave this as step **1** in the **Step Number** selection box as shown in the following screenshot:



30. Click on **Next**.

> Here, we are presented with the **Step 2. Specify Step Criteria** page.

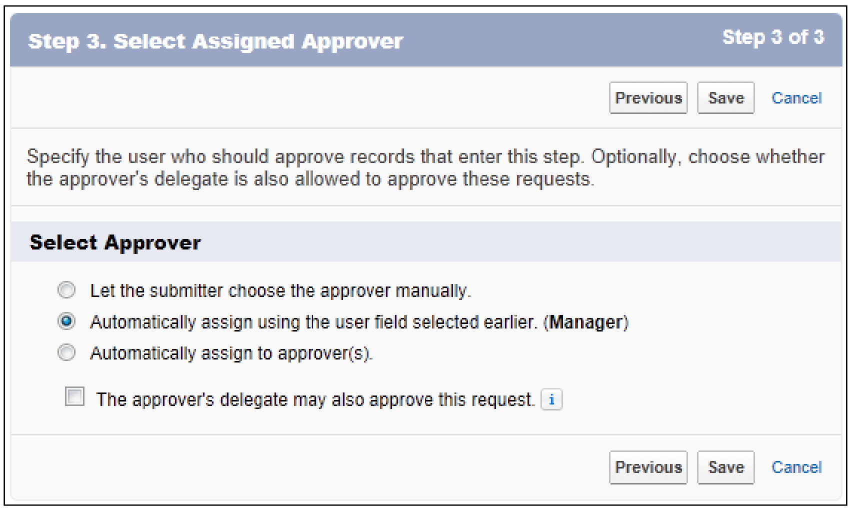
31. Choose the **All records should enter this step** option in the **Specify Step Criteria** section as shown in the following screenshot:



32. Click on **Next**.

> Here, we are presented with the **Step 3. Select Assigned Approver** page.

33. Choose the **Automatically assign using the user field selected earlier. (Manager)** option in the **Select Approver** section as shown in the following screenshot:



34. Click on **Save**.

> Here, we are presented with a **What Would You Like To Do Now?** page to allow the creation of optional workflow actions. The actions will fire based on whether the approval process is approved or rejected.

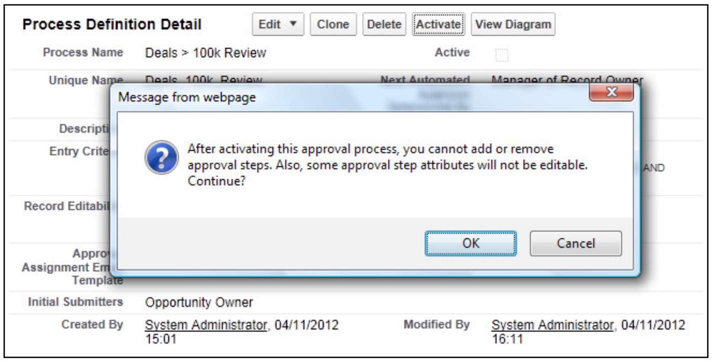
35. Choose the option **No, I'll do this later. Take me to the approval process detail page to review what I've just created** as shown in the following screenshot:

36. Click on **Go!**.

> Here, we are presented with the **Process Definition Detail** page where we can activate the approval process.

37. Click on **Activate**.

Here, we are presented with a confirmation dialog as shown in the following screenshot:

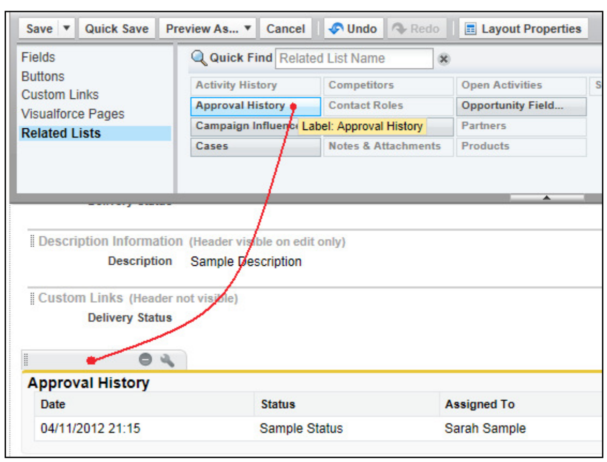


38. Click on **OK**.

After creating the approval process, we need to add the approval history-related list to the appropriate opportunity page layouts (this is to allow the **Submit for Approval** button to be presented to users).

To add the approval history-related list to the opportunity page layout, carry out the following steps:

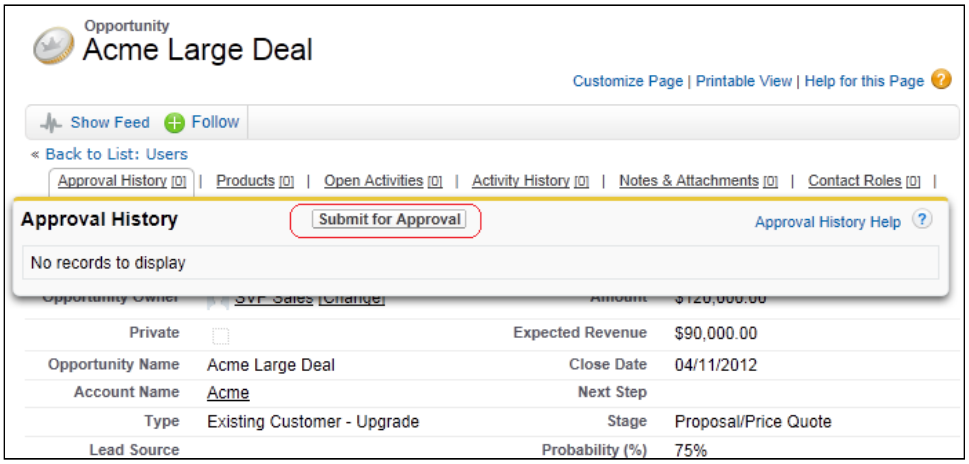
1. Navigate to the opportunity page layout setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Page Layouts**.
2. Choose the required opportunity page layout and click on **Edit**.
3. Click on the **Related Lists** section (at the top-left side of the page).
4. Drag the **Approval History** section onto the page as shown in the following screenshot:



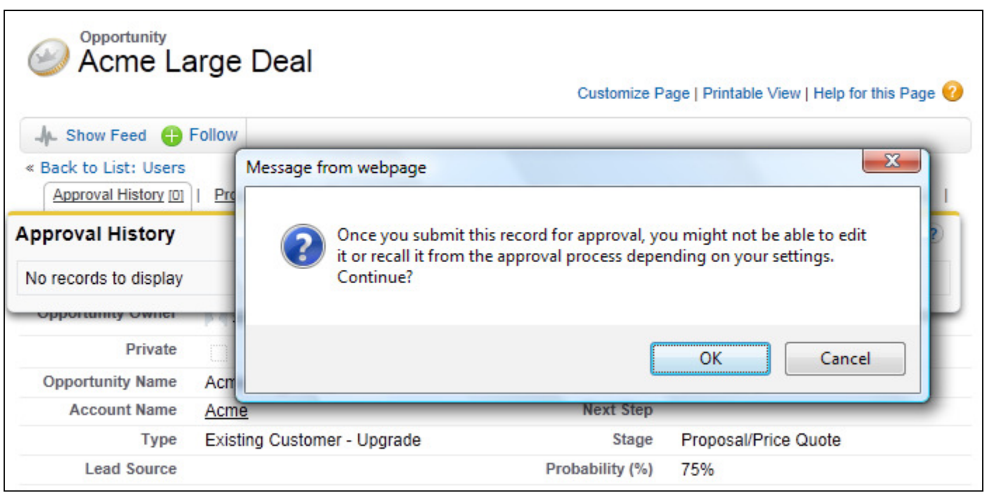
5. Click on **Save**.

How it works...

When the user has an opportunity that meets the criteria for the approval process, they may click on the **Submit for Approval** button on the **Approval History** list as shown in the following screenshot:

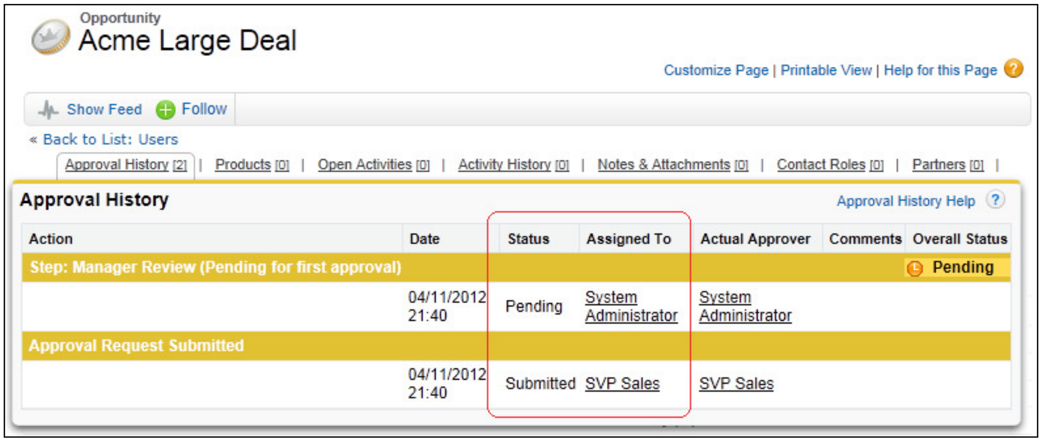


Upon clicking on the **Submit for Approval** button, the user is then presented with the following confirmation:

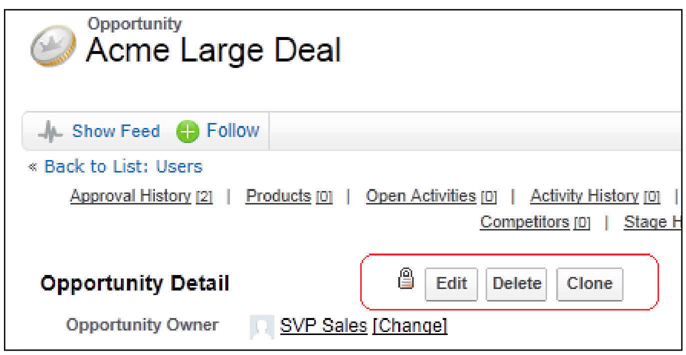


When they click on **OK**, the opportunity record is sent into the approval process and their specified manager receives an e-mail using the e-mail template that was created in the *Creating an e-mail template for use with approval assignments* recipe.

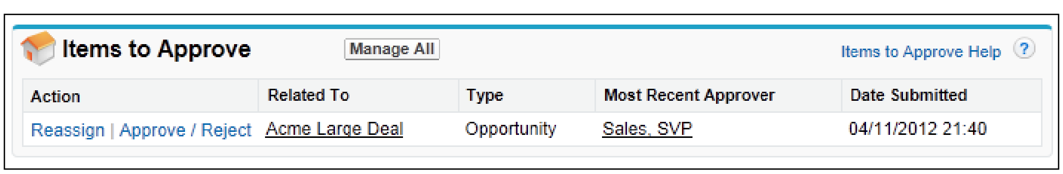
The approval history-related list on the opportunity now appears with the following status:



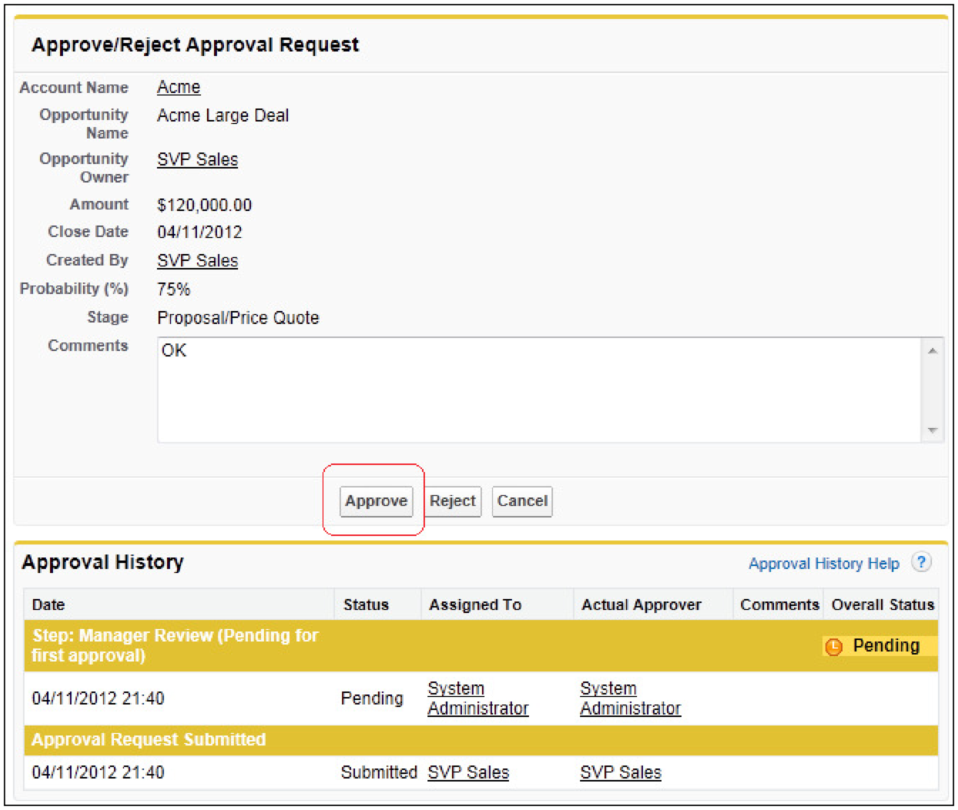
The opportunity record remains locked while the status is **Pending** and will display the locked icon (as shown in the following screenshot) until the record owner's manager approves (or rejects) the the approval process.



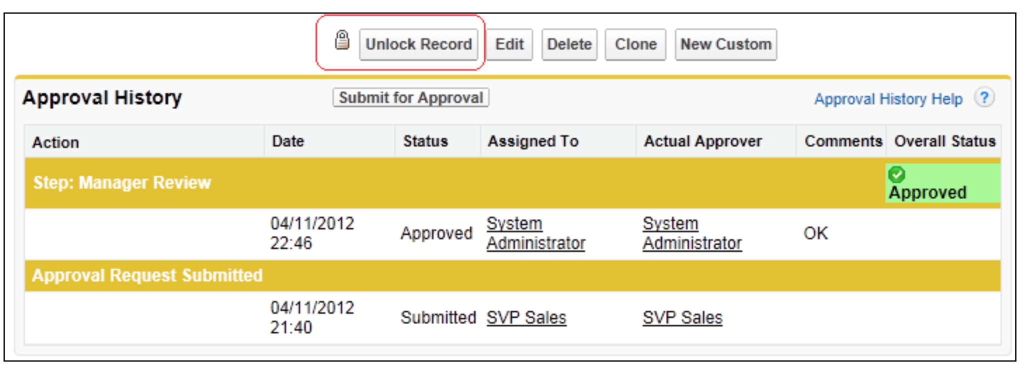
The user's manager can access the **Items To Approve** related list from their home page to get an instant view of all the approval requests they need, to approve or reject as shown:



By clicking the **Approve / Reject** link, the user is presented with the **Approve/Reject Approval Request** screen as shown in the following screenshot:



Finally, upon clicking the **Approve** button, the approval history for the opportunity record is marked as **Approved** and the approver can then unlock the record as shown:



7

Productivity Tools for Superusers and Advanced Administration

In this chapter, we will cover the following recipes:

- Converting a 15-digit Salesforce opportunity ID to 18 digits using a formula field
- Generating help pages for custom objects using Visualforce and a PDF document
- Building a mass delete opportunity button using a custom list button

Introduction

In this chapter, we will look at some useful ways in which to increase productivity for administrators and advanced users (or superusers) within Salesforce CRM. There are aspects of routine Salesforce functionality and maintenance that often require manual effort such as linking Salesforce records to external systems, presenting help pages, and deleting multiple records, recipes for which will be covered.

Productivity tools for administration include the mechanism to export records from Salesforce and be able to link those records using tools such as Microsoft Excel. Here, we look at a mechanism to ensure that the IDs in Salesforce can be handled correctly within these types of external system.

We will also look at a way to build help pages within Salesforce to display context-specific help information that increases users' productivity and frees administrators to concentrate on their tasks in hand.

Finally, we will build a routine that extends the Salesforce platform by providing a way to delete multiple opportunity records by selecting the appropriate records from the related accounts page.

> Allowing users to delete multiple records may not be suitable for all organizations. However, this advanced recipe allows administrators to delete unwanted records and may also be suitable for organizations that allow superusers to delete records.

Converting a 15-digit Salesforce opportunity ID to 18 digits using a formula field

Internal record IDs in salesforce.com, as found in all custom and standard objects such as the opportunity, account, contact, and so on contain 15 character text-based values.

The text values can be described as a base-62 number as each of the individual 15 characters can be either a numeric digit (in the range 0-9), a lowercase letter (in the range a-z), or an uppercase letter (in the range A-Z).

These 15 character salesforce.com values are therefore case-sensitive, since there can be two unique IDs which owe their uniqueness to the fact that they have a character(s) that differs only in case. For example, 100000000000ABC is different to 100000000000abc.

However, there are applications such as Microsoft Excel, which are not case-sensitive, that do not recognize the difference between the ID 100000000000ABC and the ID 100000000000abc. This results in features such as the Excel function `VLOOKUP`, failing to match correct records from an exported salesforce.com report.

Salesforce.com realized the potential problem where the case of the source lookup value is not processed correctly within external applications that are not *case-aware*.

To provide better compatibility between salesforce.com and tools such as MS Excel, MS Access, and SQL Server salesforce.com has established an 18-character case-insensitive ID that is formed by adding a suffix to the 15 character ID, a method that respects the case-insensitive nature of these tools.

In this recipe, we will convert a 15-character case-sensitive ID to an 18-character case-insensitive ID.

How to do it...

Carry out the following steps in order to convert a 15-digit Salesforce opportunity ID to 18 digits:

1. Navigate to the opportunity customization setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Fields**.

   > Scroll down to the **Opportunity Custom Fields & Relationships** section.

2. Click on **New**.

   > Here, we are presented with the **Step 1. Choose the field type** page.

3. Select the **Formula** option.
4. Click on **Next**.

   > Here, we are presented with the **Step 2. Choose output type** page.

5. Type `Opportunity ID (18 Chars)` in the **Field Label** textbox.
6. Click on the **Field Name** textbox. When clicking out of the **Field Label** textbox the field name is automatically filled with the value **Opportunity_ID_18_Chars**.
7. Set the **Formula Return Type** value as **Text**.
8. Click on **Next**.

   > Here, we are presented with the **Step 3: Enter formula** page.

9. Paste the following code:

   ```
   CASESAFEID( Id )
   ```

10. In the **Blank Field Handling** section, select the option **Treat blank fields as blanks**.

11. Click on **Next**.

> Here, we are presented with the **Step 4. Establish field-level security** page.

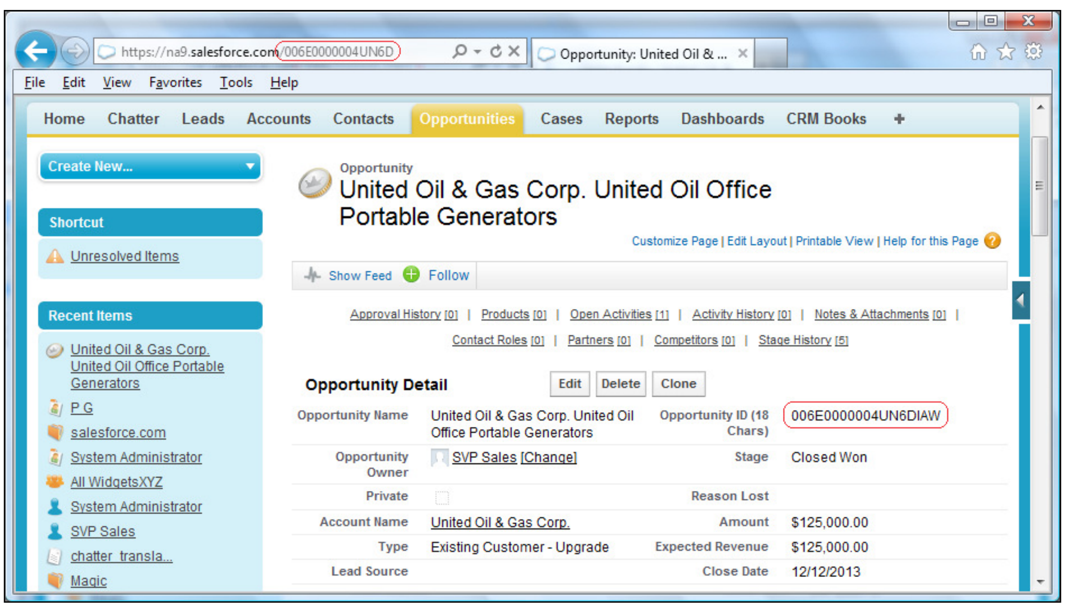
12. Select the profiles to which you want to grant read access to this field via field-level security. The field will be hidden from all profiles if you do not add it to field-level security.
13. Click on **Next**.

> Here, we are presented with the **Step 5. Add to page layouts** page.

14. Select the page layouts that should include this field. The field will be added as the last field in the first two-column section of the selected page layouts.
15. Finally, click on **Save**.

How it works...

The formula field appears on the opportunity detail page and displays the 18-character ID. You can see how this ID compares to the 15-character ID that is generated within the web URL as shown in the first red-circled section of the following screenshot:



Generating help pages for custom objects using Visualforce and a PDF document

In this recipe, we will step through the process of adding a custom PDF document for the purpose of showing help information. This help information will then be accessible from appropriate custom object records to provide for context-specific help information.

Getting ready

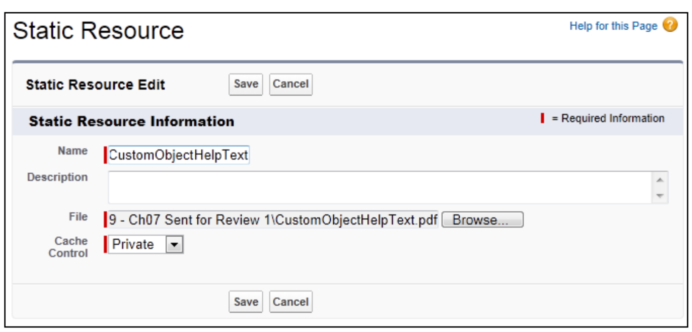
Create or retrieve a PDF document that contains the required help information. Here we have an example file that is called `HelpText.pdf`.

Upload the PDF document into Salesforce as a static resource by carrying out the following steps:

1. Create or source a suitable PDF to be used to show help information.
2. Navigate to the **Static Resources** setup page, by clicking along the following path: **Your Name** | **Setup** | **Develop** | **Static Resources**.
3. Click on **New**.
4. Enter the name of the static resource in the **Name** field. For this recipe, type the text `CustomObjectHelpText`.
5. In the **File uploader** control, click on the **Browse...** button to select the document to upload from your computer. For this recipe, choose the document identified in step 1.

> Ignore the **Cache Control** pick list selection and leave it as the default **Private** value (**Cache Control** is only relevant to static resources used in the Force.com sites).

6. The **Static Resources** setup screen appears as shown in the following screenshot:



7. Click on **Save**.

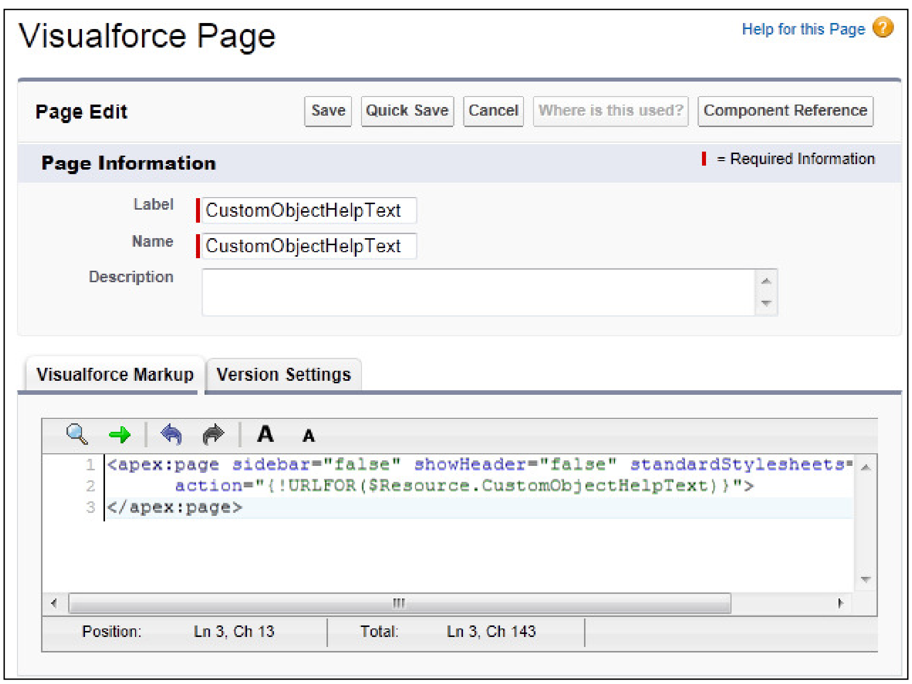
Now that we have a PDF document stored within Salesforce, we can reference the document from within a Visualforce page. This Visualforce page can then be accessed from the link called **Help for this Page link**, which is displayed on a custom object record detail page.

How to do it...

Carry out the following steps in order to generate a help page for custom objects using Visualforce and a PDF document:

1. Navigate to the Visualforce pages setup screen, by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `CustomObjectHelpText` in the **Label** field.
4. Accept the default **CustomObjectHelpText** in the **Name** field.
5. Paste the following code:

```
<apex:page sidebar="false" showHeader="false"
standardStylesheets="false"
      action="{!URLFOR($Resource.CustomObjectHelpText)}">
</apex:page>
```

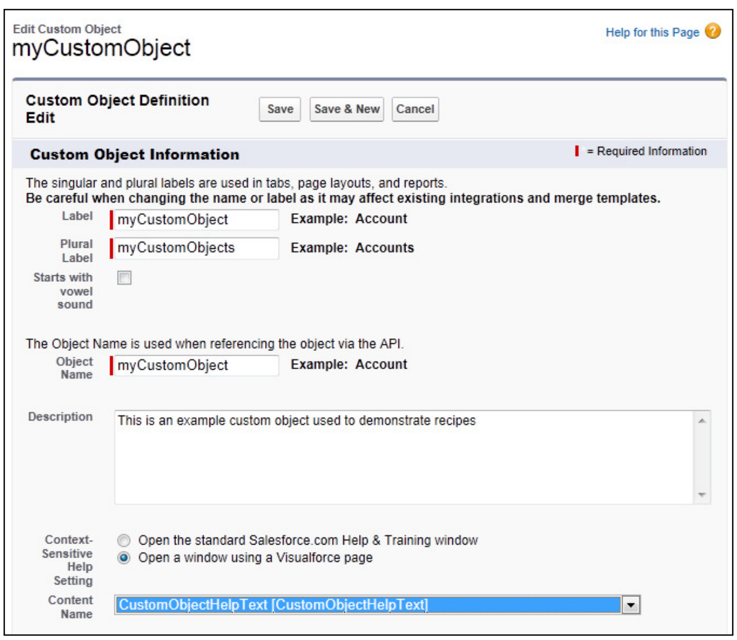


6. Click on **Save**.

7. Now set the security for the required profiles in your organization by going to **Your Name** | **Setup** | **Develop** | **Pages**.
8. Locate the row for the Visualforce page **CustomObjectHelpText** and click on the **Security** link.
9. Set the security for the required profiles.
10. Now, set the Visualforce page as the resource behind the **Help for this Page** link by navigating to your custom object customization setup page.
11. In this recipe, we have a custom object named **myCustomObject**. Here we proceed by going to **Your Name** | **Setup** | **Create** | **Objects** | **myCustomObject**.
12. Click on **Edit**.

> Here, we are presented with the **Custom Object Definition Edit** page.

13. In the **Context-Sensitive Help Setting** option, select the **Open a window using a Visualforce page** choice.
14. In the **Content Name** pick list, select the Visualforce page called **CustomObjectHelpText** that we created previously as shown in the following screenshot:

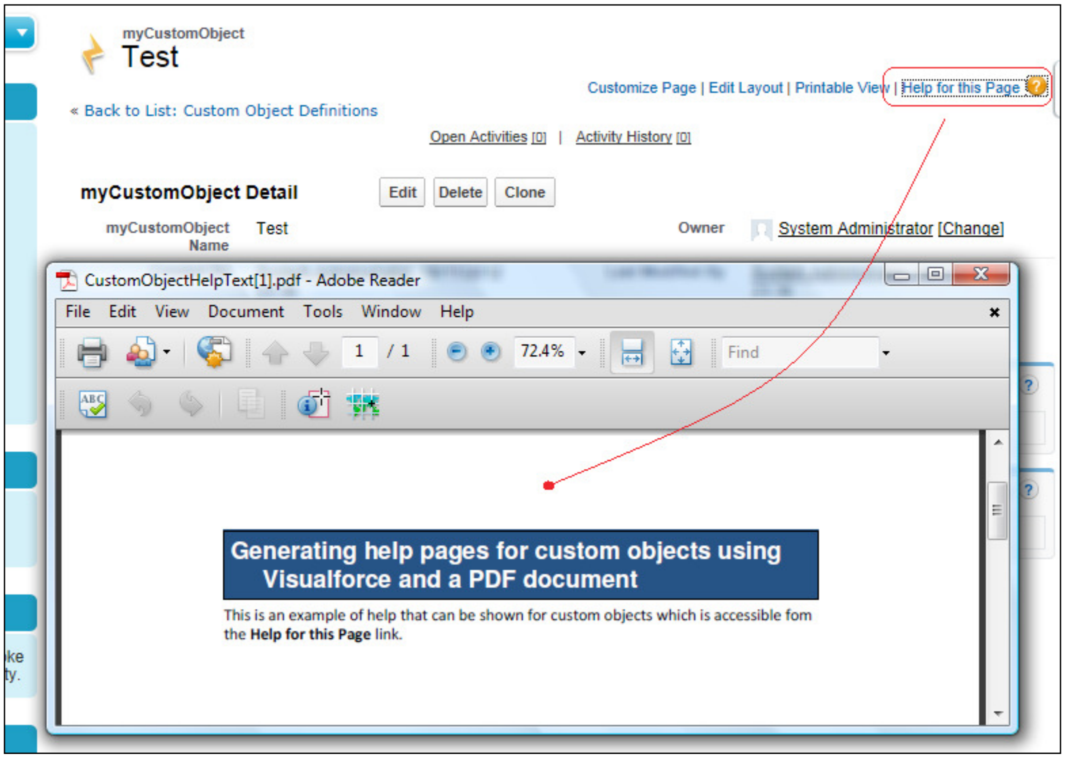


15. Finally, click on **Save**.

How it works...

From within the detail page of a custom record, we are presented with a link (at the top right-hand side of the page) called **Help for this Page** as shown in the upcoming screenshot.

When clicking on the **Help for this Page** link, the PDF document that we uploaded previously is opened in a new window as shown in the following screenshot:



Building a mass delete opportunity button using a custom list button

In this recipe, we will build a custom list button for the deletion of opportunities that we can add to the opportunity-related list on our account detail page.

This allows us to select multiple opportunity records and delete them with a single click.

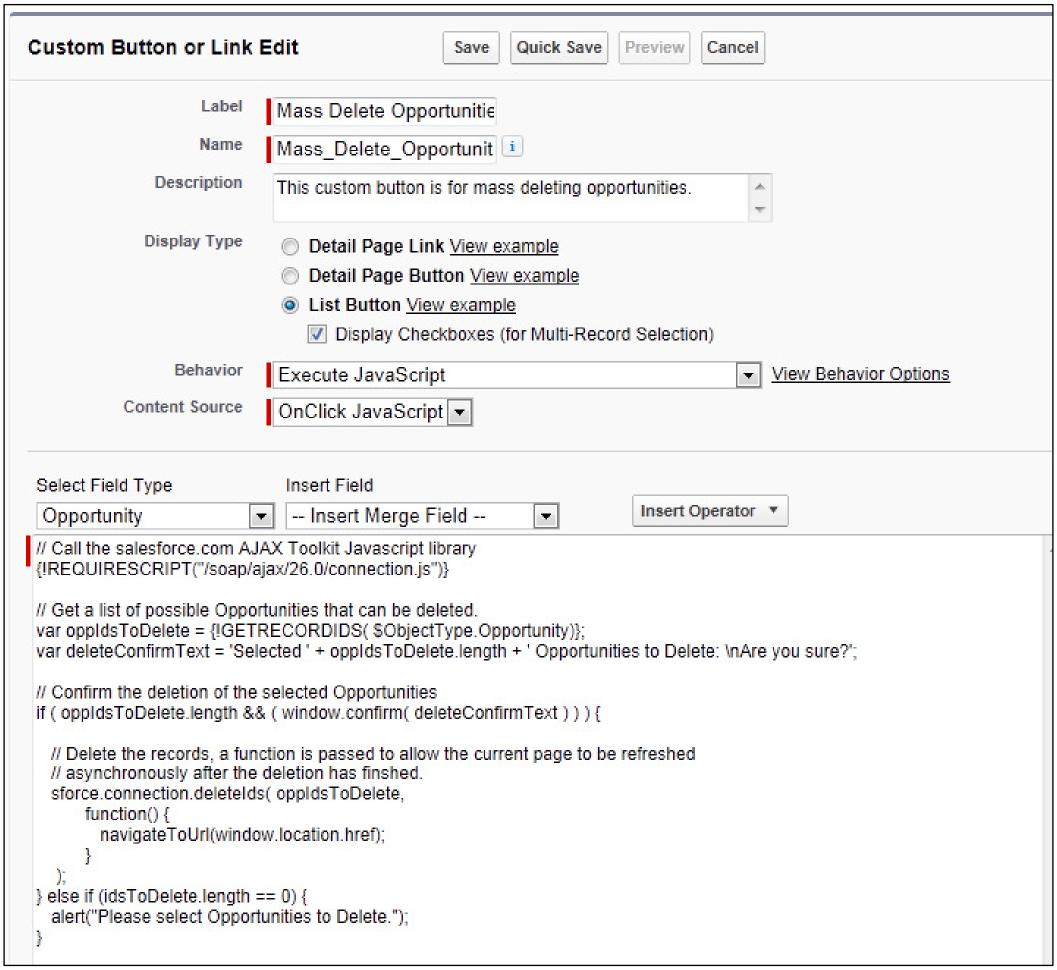
How to do it...

Carry out the following steps in order to build a mass delete opportunity button using a custom list button:

1. Navigate to the opportunity **Buttons and Links** setup page, by going to **Your Name** | **Setup** | **Customize** | **Opportunities** | **Buttons and Links**.
2. Click on **New**.
3. Type `Mass Delete Opportunities` in the **Label** field.
4. Accept the default value **Mass_Delete_Opportunities** that is automatically set in the **Name** field.
5. Optionally, set the **Description** field to `This custom button is for mass deleting opportunities`.
6. In the **Display Type** field, select the option **List Button**.
7. Check the box **Display Checkboxes (for Multi-Record Selection)**.
8. Set the **Behavior** pick list to **Execute JavaScript**.
9. Set the **Content Source** pick list to **OnClick JavaScript**.
10. Enter the following code:

```
// Call the salesforce.com AJAX Toolkit Javascript library
{!REQUIRESCRIPT("/soap/ajax/26.0/connection.js")}

// Get a list of possible Opportunities that can be deleted.
var oppIdsToDelete = {!GETRECORDIDS( $ObjectType.Opportunity)};
var deleteConfirmText = 'Selected ' + oppIdsToDelete.length + '
Opportunities to Delete: \nAre you sure?';

// Confirm the deletion of the selected Opportunities
if ( oppIdsToDelete.length && ( window.confirm( deleteConfirmText
) ) ) {

   // Delete the records, a function is passed to allow the
   //current page to be refreshed
   // asynchronously after the deletion has finished.
   sforce.connection.deleteIds( oppIdsToDelete,
     function() {
       top.location.replace( '/{!Account.Id}' );
     }
    );
 } else if (oppIdsToDelete.length == 0) {
   alert('Please select Opportunities to Delete.');
}
```

11. Optionally, click on **Check Syntax**.

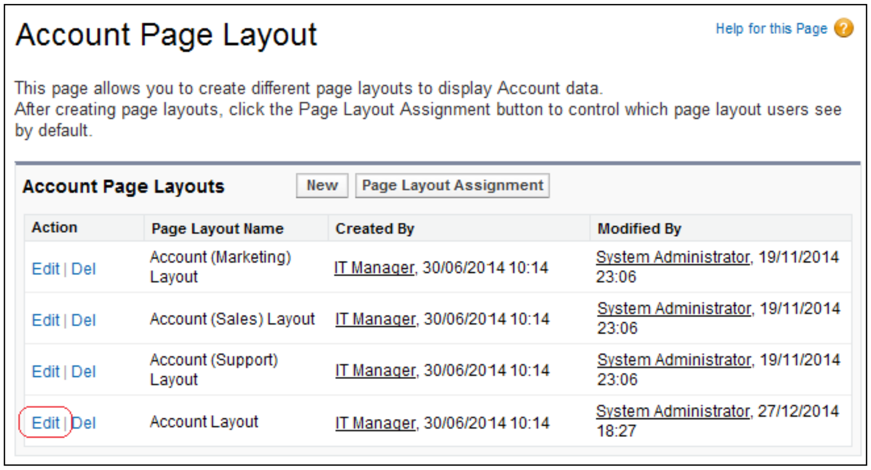


12. Click on **Save**.

> Now we will add the custom button to the opportunities-related list on the account page layout.

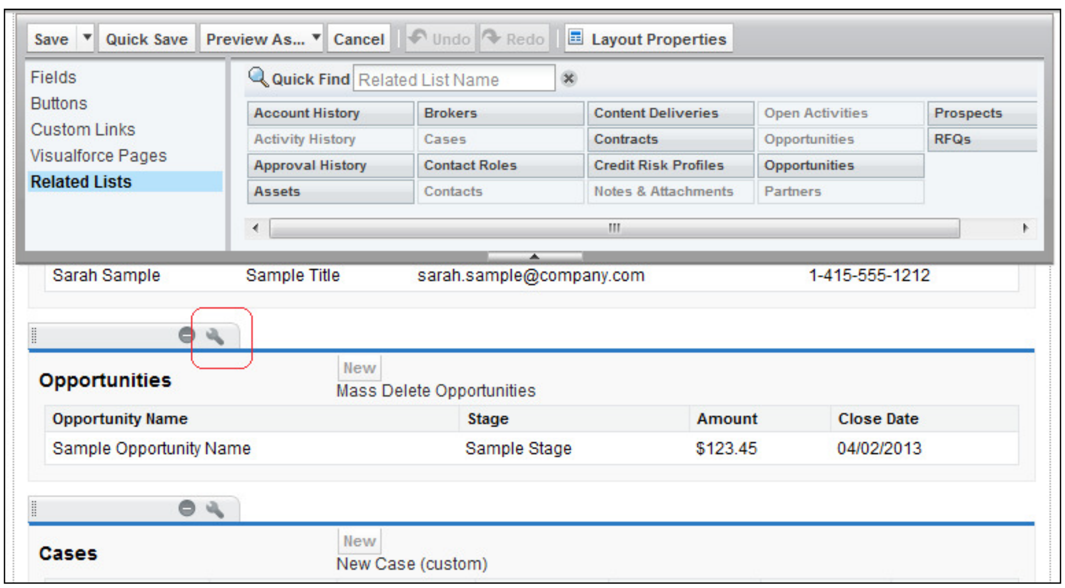
13. Navigate to the account **Page Layout** setup page, by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Page Layouts**.

14. Select an appropriate page layout and click on **Edit** (an example is shown in the following screenshot):

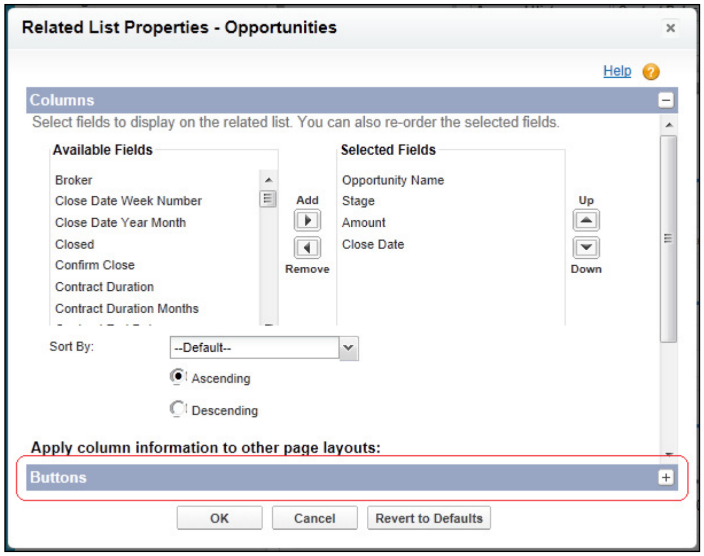


15. Click on the **Related Lists** section (in the top-left area of the setup page).

16. Access the opportunities-related list and click on the **Related List Properties** icon as shown in the following screenshot:

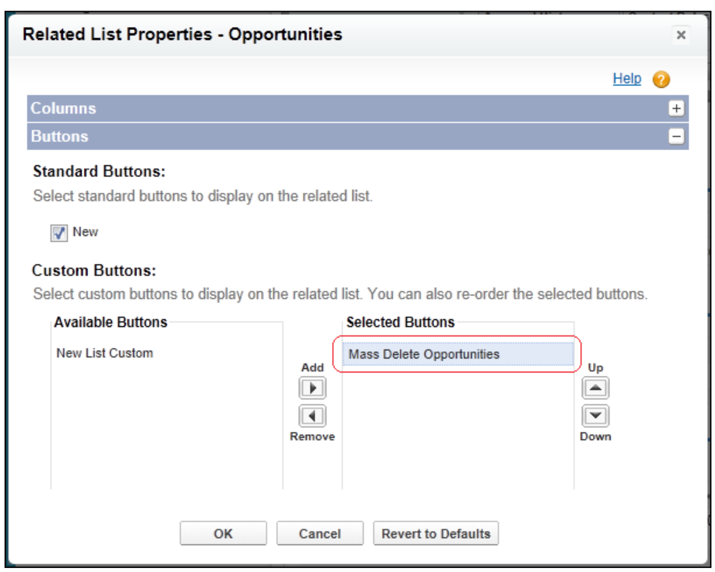


17. Within the **Related List Properties** window section, scroll down to the bottom section marked **Buttons** as shown in the following screenshot:



18. Click on the **+** button at the right-hand side of the **Buttons** section.

> Here, the following button-select area of the page is presented.



19. Select the **Mass Delete Opportunities** button from the left-hand side, available buttons section and move it to the right-hand side, selected button section as shown in the preceding screenshot.
20. Click on **OK**.

> Here, we are directed back to the **Account Layout** page that was selected for edit previously.

21. Click on **Save**.

How it works...

Whenever appropriate users navigate to an account record, they are able to access a **Mass Delete Opportunities** button on the opportunity-related lists.

By selectively choosing the opportunity records that are to be deleted, using the associated checkbox, and then clicking the **Mass Delete Opportunities** button, all of the selected records are removed with one click.

The solution uses a custom button that is set to handle multiple records and executes JavaScript.

The custom button works by first invoking the Salesforce AJAX Toolkit which is a JavaScript wrapper around the Salesforce API. This is invoked with the code `{!REQUIRESCRIPT("/soap/ajax/26.0/connection.js")}`.

> The AJAX Toolkit is available for any organization that has API access.

The JavaScript code works by retrieving the multiple opportunity records that the user has selected for deletion. This is performed with the code `{!GETRECORDIDS( $ObjectType.Opportunity)};`.

> `GETRECORDIDS` is a Salesforce function that returns an array of strings in the form of record IDs for the selected records in a list, such as a list view or related list. The syntax is `{!GETRECORDIDS(object_type)}` where `object_type` is the reference to the custom or standard object for the records you want to retrieve.

The JavaScript code then checks whether there are records to be deleted and if so prompts for confirmation from the user to proceed. This is executed with the code `if ( oppIdsToDelete.length && ( window.confirm( deleteConfirmText ) ) )`.

> The `window.confirm()` method is a standard JavaScript method. The `confirm()` method displays a dialog box with a specified message, along with an **OK** button and a **Cancel** button. This method returns true if the visitor clicked on **OK**, and `false` otherwise.

When the user clicks on **OK**, the JavaScript code invokes the Salesforce AJAX Toolkit function that deletes the records that have been selected. This is performed with the code `sforce.connection.deleteIds( oppIdsToDelete, .`

> The AJAX Toolkit allows you to issue **synchronous** or **asynchronous** calls. Asynchronous calls allow the client-side process to continue, waiting for a call back from the server. To issue an asynchronous call, you must include an additional parameter with the API call, referred to as a **callback** function. Once the result is ready, the server invokes the callback method with the result.

In our code, we are issuing an asynchronous call and using a callback function that causes the account detail page to be refreshed. This clears the opportunity-related list of the records that were deleted and is executed with the following code:

```
function() {
        top.location.replace( '/{!Account.Id}' );
       }
```

There's more...

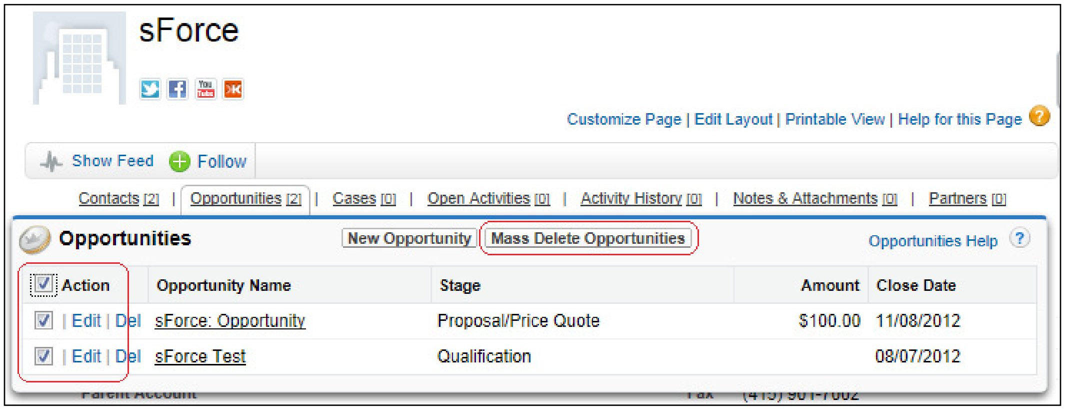
Carry out the following steps in order to mass delete opportunities:

1. Select an example account record that has more than one opportunity record associated.
2. Hover over (or navigate) to the opportunity-related list section.

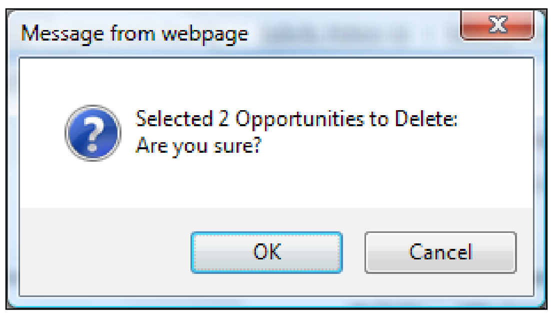
> Here, we now have the custom **Mass Delete Opportunities** button and an **Action** select all checkbox along with checkboxes to the left of each of the opportunities.

3. Set the checkbox for all of the opportunity records that you wish to delete.

4. Click on the **Mass Delete Opportunities** button as shown in the following screenshot:



5. Upon clicking on the **Mass Delete Opportunities** button, we are presented with a confirmation message box as shown in the following screenshot:



6. Click on **OK**.

[The records will then be deleted and the account detail page is refreshed to confirm that the selected records are no longer present.]

8

Configuring and Installing Salesforce for Outlook E-mail Integration

In this chapter, we will cover the following recipes:

- Enabling the Email to Salesforce feature in Salesforce CRM
- Obtaining system requirements information for Salesforce for Outlook
- Setting up Salesforce for Outlook configurations within Salesforce CRM
- Installing the Salesforce for Outlook software on a local machine
- Configuring the Salesforce for Outlook software on a local machine

Introduction

E-mail applications often allow a wealth of valuable information about users' business contacts and the communications with those contacts to be stored and managed by Salesforce CRM users.

Having visibility of this information within Salesforce is highly desirable by the users of Salesforce CRM. Additionally, users often want to continue using their e-mail applications alongside Salesforce and to feel confident that the two systems are automatically kept in sync.

Enabling the Email to Salesforce feature in Salesforce CRM

Email to Salesforce is a feature within Salesforce CRM that lets you assign e-mails to Salesforce records when sending messages from your external e-mail account, such as Outlook, Google Mail, and so on. Here, you can assign the e-mail to leads, contacts, opportunities, and other records in Salesforce.

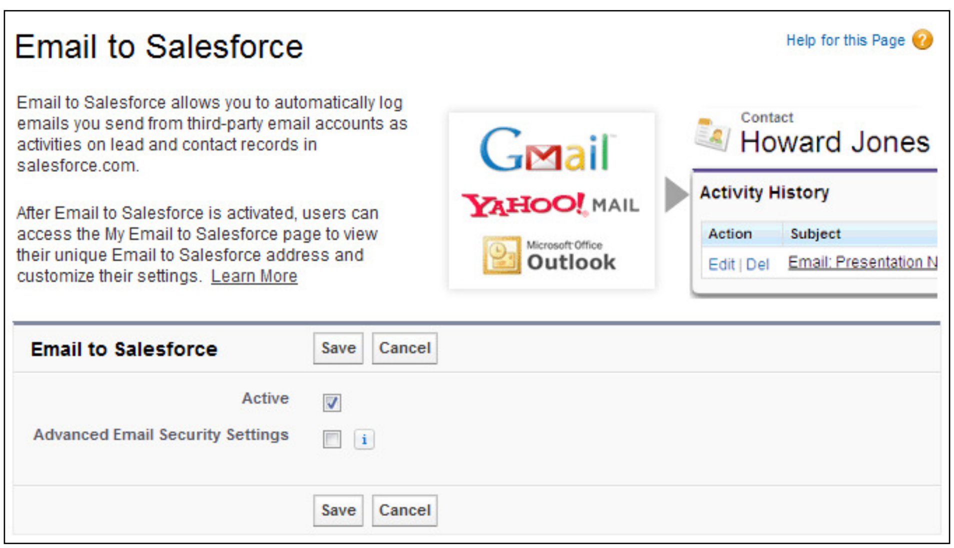
When creating, forwarding, or replying to an e-mail, you simply enter your Email to Salesforce address in the **BCC** field or any recipient field.

The Salesforce system then receives a copy of the e-mail and, depending on your configuration, adds it to either the **Activity History** related list of the matched record or to the **My Unresolved Items** page, where you can then manually assign it.

How to do it...

Carry out the following steps to enable the Email to Salesforce feature in Salesforce CRM:

1. Navigate to the Email to Salesforce setup page by going to **Your Name** | **Setup** | **Email Administration** | **Email to Salesforce**.
2. Click on **Edit**.
3. Select the **Active** checkbox as shown in the following screenshot:



4. Optionally, select the **Advanced Email Security Settings** checkbox to configure **Email to Salesforce** to verify the legitimacy of the sending server before processing a message.

> Before enabling this option, make sure that your users' e-mail domains support at least one of the following protocols: SPF, Sender ID, or DomainKeys.
>
> When this option is enabled, the Salesforce system uses these protocols to verify the legitimacy of the e-mail sender's server. If the server passes at least one protocol and does not fail any, the system processes the e-mail. If the server fails a protocol or does not support any of the protocols, then the Salesforce system does not process the e-mail.

5. Click on **Save**.

There's more...

After enabling Email to Salesforce, Salesforce creates a unique Email to Salesforce address for each user.

Users can view their unique Email to Salesforce address and customize their settings by carrying out the following steps:

1. Navigate to the Email to Salesforce setup page by going to **Your Name** | **Setup** | **Email** | **My Email to Salesforce**.
2. In **My Acceptable Email Addresses**, enter any e-mail addresses that you use to send e-mail (you can add multiple e-mail addresses separated by commas).

> Your Email to Salesforce address only accepts e-mail from addresses listed here. If you do not list any e-mail addresses, the e-mail sent to your Email to Salesforce address will not be associated to any records.

3. Choose whether you want e-mails to be sent to **My Unresolved Items** so that you can manually assign them or whether you want the Salesforce CRM system to try to automatically assign them.
4. Select whether to add e-mails to opportunities, leads, and/or contacts.

5. If leads or contacts are selected, choose how **Email to Salesforce** is to handle lead or contact e-mails that match duplicate records. **Email to Salesforce** can add the e-mail activity to either: all matching records; to the oldest duplicate record or; to the record that has greatest number of activities.
6. If leads or contacts are selected, you can select the option **If no matching records are found, create a task and send it to My Unresolved Items** to route the e-mail to **My Unresolved Items** if no matching records are found.

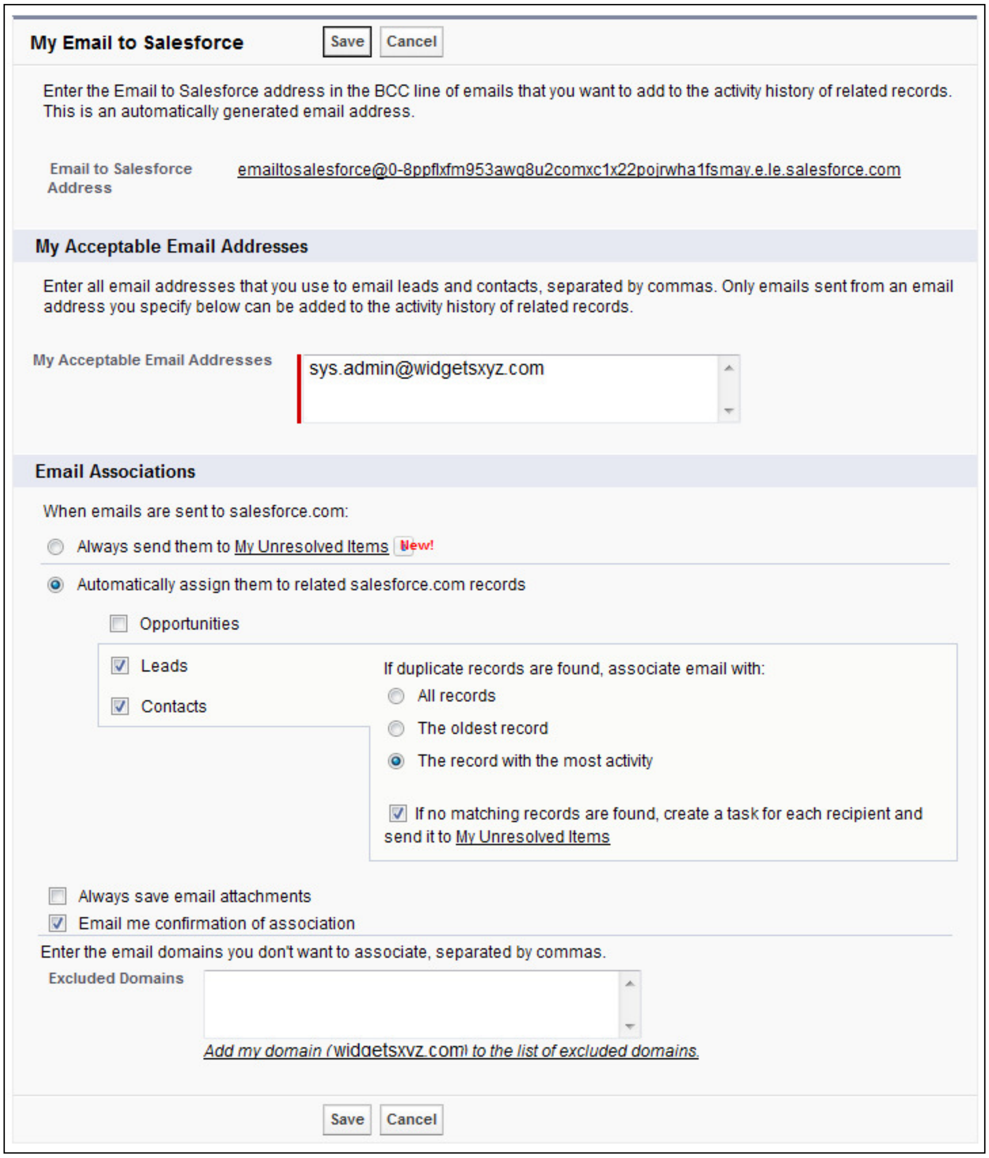
> If this checkbox is not selected and Salesforce cannot identify the To or From e-mail addresses, the e-mail will not be saved in Salesforce.

7. Select **Always save email attachments** to save attachments on e-mails sent to Salesforce.

> The maximum size for an attachment is 5 MB when attached directly to the related list, and the size limit for all files attached to an e-mail is 10 MB.

8. Select **Email me confirmation of association** to receive a confirmation e-mail when e-mails are successfully associated.
9. In **Excluded Domains**, enter any e-mail domain you want to exclude from automatic association. You can add multiple e-mail addresses separated by commas.

10. Click on **Save** as shown in the following screenshot:



Obtaining system requirements information for Salesforce for Outlook

Many users of Salesforce also have Microsoft Outlook as their e-mail application and in order to link Outlook data with Salesforce, salesforce.com has developed **Salesforce for Outlook**.

Salesforce for Outlook is an external application that installs onto the same computer as the Outlook application and it can be used to keep contact, event, and task records synced between Salesforce and Outlook.

There are a number of steps required in the setup and configuration of Salesforce for Outlook which we cover in this chapter. Before attempting to set up and install Salesforce for Outlook, you should check that the hardware and software specifications of the users' machines—that you are planning for the installation—meet the following system requirements:

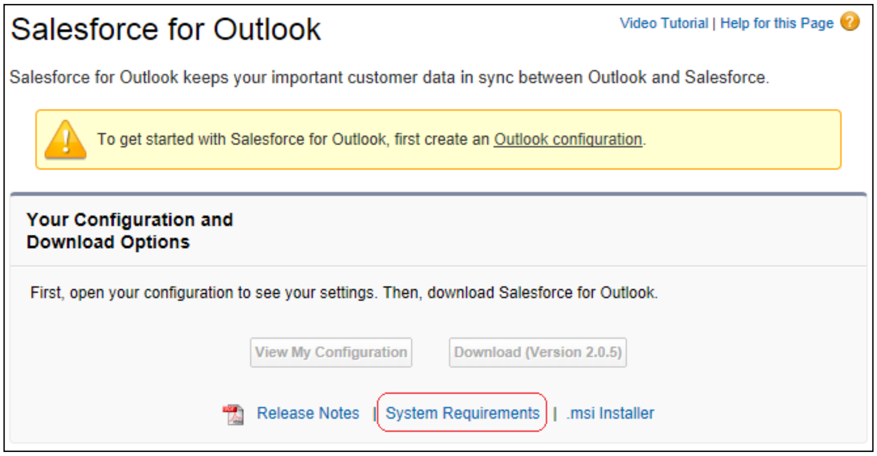
- Microsoft Outlook 2010 (64 bit and 32 bit) or 2007 on Microsoft Exchange
- Microsoft Windows 7 (64 bit and 32 bit), Windows Vista (32 bit only), or Windows XP (32 bit only) on Microsoft Exchange
- Latest versions of hotfixes for Outlook and Windows
- These specifications are correct at the time of printing but will be subject to change and you should obtain the system requirements information for Salesforce for Outlook from within Salesforce CRM.

How to do it...

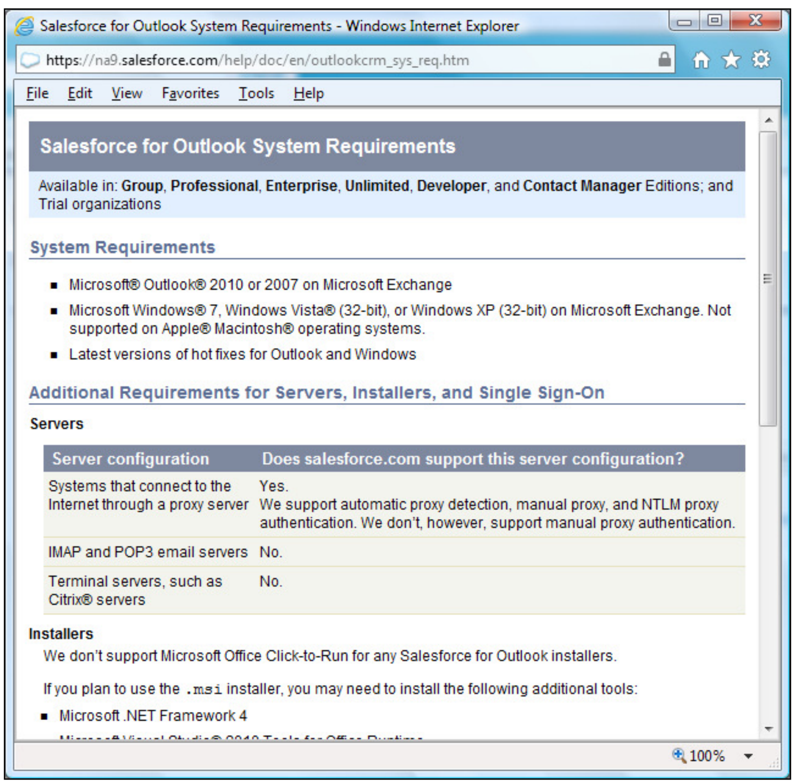
Carry out the following steps to obtain system requirements information for Salesforce for Outlook:

1. Navigate to the Salesforce for Outlook page by going to **Your Name** | **Setup** | **Desktop Integration** | **Salesforce for Outlook**.

2. Click on the **System Requirements** button as shown in the following screenshot:



3. Review the system requirements as shown in the resulting screen (this is shown in the following screenshot):



Setting up Salesforce for Outlook configurations within Salesforce CRM

Many users of Salesforce also have Microsoft Outlook as their e-mail application and in order to link Outlook data with Salesforce, salesforce.com has developed Salesforce for Outlook.

Salesforce for Outlook is an external application that installs onto the same computer as the Outlook application and it can be used to keep contact, event, and task records synced between Salesforce and Outlook.

A Salesforce for Outlook configuration comprises the settings and parameters that determine which data users can sync between Outlook and Salesforce CRM.

Salesforce for Outlook configuration also allows you to set up configurations that allow users to edit some of their own settings.

Within the Salesforce for Outlook configuration, you can define which items are allowed to sync, which direction the data flows between Outlook and Salesforce, and what happens when data conflicts arise.

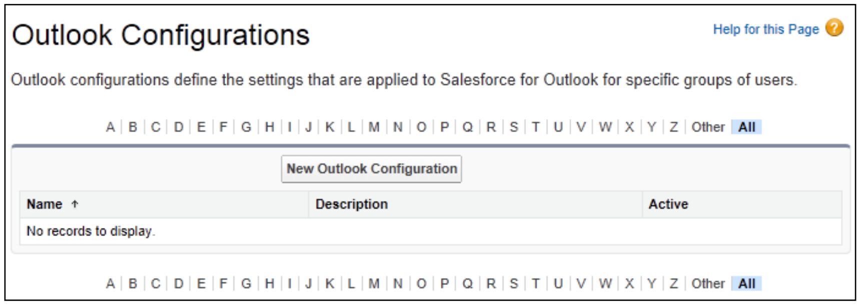
> Salesforce for Outlook is available for the following editions: Group, Professional, Enterprise, Unlimited, Developer, and Contact Manager Editions, plus Trial organizations.

You can create multiple configurations to cater for the needs of various types of users. For example, a sales team might want to sync contacts, tasks, and events, while a senior manager might only be concerned with syncing events.

How to do it...

Carry out the following steps to set up Salesforce for Outlook configurations within Salesforce CRM:

1. Navigate to the new **Outlook Configuration** page by going to **Your Name** | **Setup** | **Desktop Administration** | **Outlook Configurations**. This page is shown in the following screenshot:



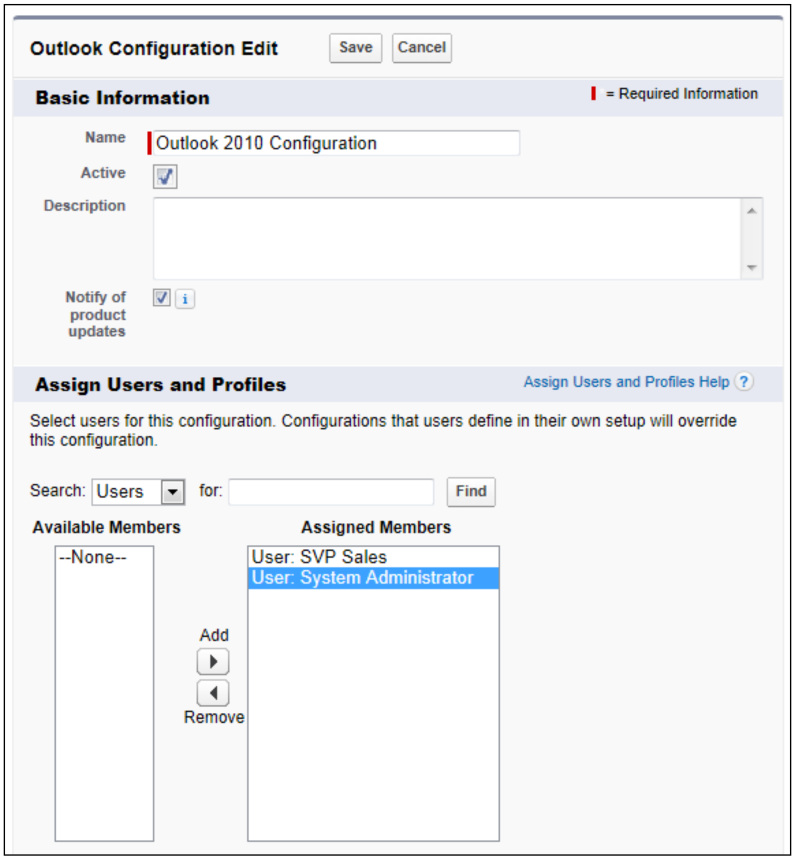
2. Click on **New Outlook Configuration**.

> Here, we are presented with the **New Outlook Configuration** page. Complete the following details in the **Basic Information** section as shown in the upcoming screenshot.

3. Enter the name of the Outlook configuration in the **Name** field. For this recipe, type the text `Outlook 2010 Configuration`.
4. Check the **Active** checkbox.
5. Optionally, enter a description in the **Description** textbox.
6. Leave the checkbox **Notify of product updates** checked. When an updated version of Salesforce for Outlook is available, users will see a system tray notification, which they can then click on to download the latest version. Users are alerted every 14 days until they either download the new version or hide system tray alerts.

> After you define basic information for a Salesforce for Outlook configuration, you select the Salesforce for Outlook users that the configuration's settings will apply to. Complete the following details in the **Assign Users and Profiles** section as shown in the upcoming screenshot.
>
> If you assign a user who is already assigned to a different configuration, that user is removed from the other configuration and assigned to this one. If an assigned user is also part of a profile assigned to another active configuration, the configuration that lists the user directly is used.

7. Select the users or profiles for this configuration:



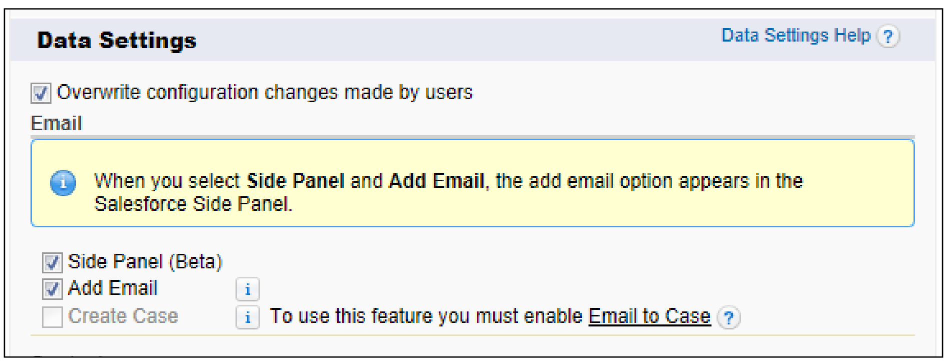
Data settings control the sync behavior and e-mail functionality of Salesforce for Outlook. You can define the following settings using the fields in the **Data Settings** section in a Salesforce for Outlook configuration. Complete the following details in the **Data Settings** section as shown in the upcoming screenshot.

8. Check the **Overwrite configuration changes made by users** checkbox.
9. Check the **Side Panel (Beta)** checkbox to display the Salesforce side panel in Microsoft Outlook.

10. Check the **Add Email** checkbox. This adds the **Add Emails** and **Send and Add** buttons to Outlook, and lets your users add up to 10 e-mails simultaneously to Salesforce.

> Before activating the **Add Email** setting, you must first have enabled the Email to Salesforce feature. Email to Salesforce allows you to assign e-mails to leads, contacts, opportunities, and other records in Salesforce when e-mailing from Outlook. This is done by entering your Email to Salesforce address in the **BCC** field or any recipient field.
>
> Salesforce then receives a copy of the e-mail and, depending on your configuration, adds it either to the **Activity History** related list of the matching records or to the **My Unresolved Items** page, where you can manually assign it.

11. Optionally, set the **Create Case** checkbox depending on your setup and requirements as shown in the following screenshot:



> We now have the options to set up the syncing of contacts, events, and task records between Salesforce CRM and Microsoft Outlook. Here we can further configure settings such as sync direction, conflict behavior, field mappings, matching criteria, and modification permissions.

12. In the **Contacts** subsection, set the **Sync Direction** option as **Salesforce.com to Outlook**.
13. Here, the **Conflict Behavior** option defaults to **Salesforce.com always wins**.

14. In the **Contacts** subsection, leave **Edit Field Mappings** unchanged.

> The **Edit Field Mappings** link allows you select fields mapping between Salesforce contacts and Outlook contacts. Clicking on the link presents the mapped fields which automatically has all standard fields mapped between Salesforce and Outlook. If you have any custom fields to sync, you can add them here.

15. In the **Contacts** subsection, set the **If an Outlook contact matches multiple salesforce.com contacts, sync it with the salesforce.com contact that:** pick list option to **Has the most recent activity**.
16. In the **Contacts** subsection, check only the **Whether object is synced** checkbox present in the **Allow users to modify:** section.

> Salesforce.com should be set up as the system of record when syncing contacts with Outlook due to the issue of record ownership in the Salesforce CRM application. Here we may have one user who owns a particular contact record and another user who may also have this contact record in their Outlook. By setting the sync direction as Salesforce.com to Outlook, we can avoid duplicate contact records being added to Salesforce.

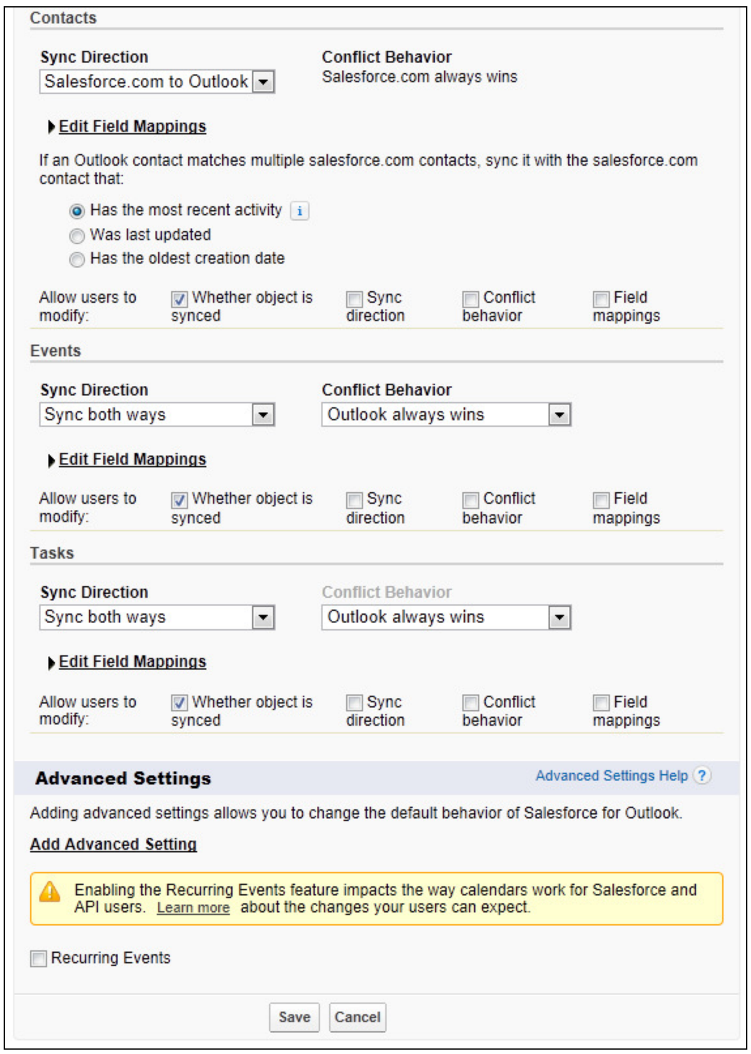
17. In the **Events** subsection, set the **Sync Direction** option as **Sync both ways**.
18. In the **Events** subsection, set **Conflict Behavior** to **Outlook always wins**.
19. In the **Events** subsection, leave **Edit Field Mappings** unchanged.

> The **Edit Field Mappings** link allows you to select fields mapping between Salesforce events and the Outlook calendar. Clicking on the link presents the mapped fields which automatically has all standard fields mapped between Salesforce and Outlook. If you have any custom fields to sync, you can add them here.

20. In the **Events** subsection, check only the **Whether object is synced** checkbox present in the **Allow users to modify:** section.
21. In the **Tasks** subsection, set the **Sync Direction** option as **Sync both ways**.
22. In the **Tasks** subsection, set **Conflict Behavior** to **Outlook always wins**.
23. In the **Tasks** subsection, leave **Edit Field Mappings** unchanged.

> The **Edit Field Mappings** link allows you to select fields mapping between Salesforce tasks and Outlook tasks. Clicking on the link presents the mapped fields which automatically has all standard fields mapped between Salesforce and Outlook. If you have any custom fields to sync, you can add them here.

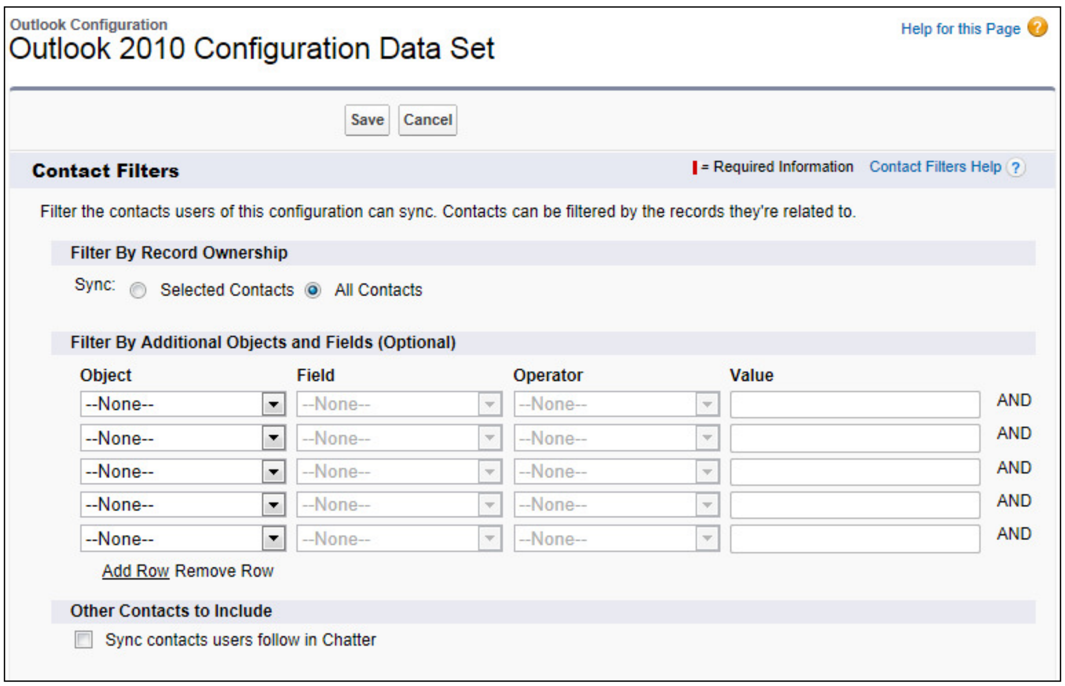
24. In the **Tasks** subsection, check only the **Whether object is synced** checkbox present in the **Allow users to modify:** section.
25. In the **Advanced Settings** section, you can leave the **HTTPTimeout**, **MaxLogFileSize**, and **MaxRetries** settings as default:



26. Click on **Save**.

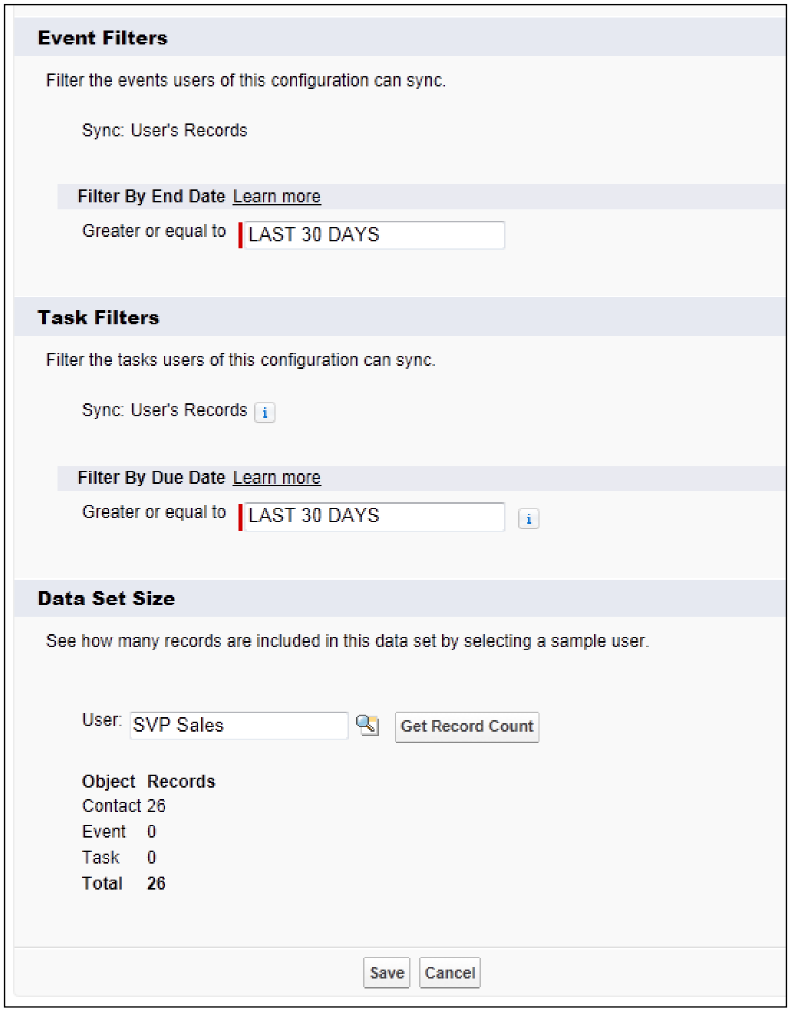
> Here, we are presented with the page to set filters and **Data Set Size**.

27. In the **Contact Filters** section, set the option for **Filter By Record Ownership** to **All Contacts**.
28. In the **Contact Filters** section, leave the pick list selections for the **Filter By Additional Objects and Fields (Optional)** as default.
29. In the **Contact Filters** section, optionally set the **Other Contacts to Include** subsection checkbox for the **Sync contacts users follow in Chatter** (as shown in the following screenshot):



30. In the **Event Filters** section, leave the **Filter By End Date** option as **LAST 30 DAYS**.

31. In the **Task Filters** section, leave the **Filter By Due Date** option as **LAST 30 DAYS** as shown in the following screenshot:



32. You can use the **Get Record Count** button to check how many records will sync for a sample Salesforce for Outlook user based on the configuration's filters. If the numbers are too high or low, you can adjust the filters and check the size again. The maximum record count is 5000 per object.

33. Click on **Save**.

Installing the Salesforce for Outlook software on a local machine

Now that we have created an Outlook Configuration, we can download and install the Salesforce for Outlook software to our local machine.

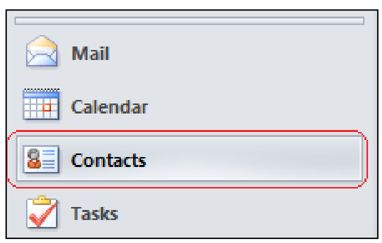
Getting ready

Before proceeding with the setup and installation, you should decide which Outlook contacts folder you want to use for the link between Salesforce and Outlook.

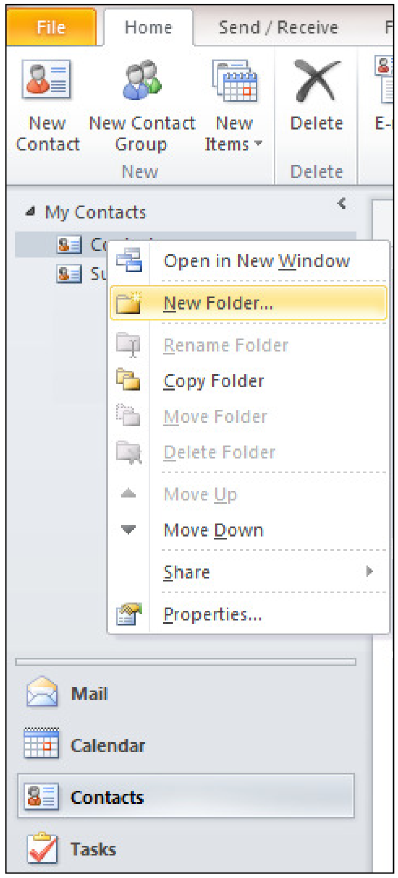
> If this, and the setup, is not considered appropriately, the sync between the two systems can result in duplicate records in either Outlook or Salesforce.

For this recipe, we are creating a contacts folder in Outlook specifically for the sync with Salesforce. We are naming the folder **Salesforce** and it is created in Outlook using the following steps:

1. Click on the **Contacts** icon in the left-hand side pane as shown in the following screenshot:



2. Click on one of the existing contact folders and then right-click on that folder to display the following options:

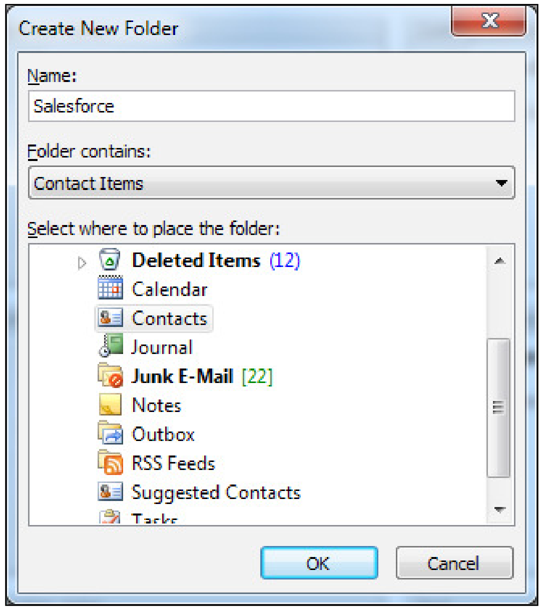


3. Click on **New Folder**.

> Here, we are presented with the **Create New Folder** dialog box as shown in the upcoming screenshot.

4. In the **Name:** text entry field, type `Salesforce`.

5. Ensure that the **Folder contains:** selection is set to **Contact Items** as shown in the following screenshot:



6. Click on **OK**.

Some further factors to consider for the setup and configuration of the **Salesforce for Outlook** software are as follows:

- **The syncing method between Salesforce and Outlook**

  Salesforce for Outlook uses the Outlook folder to limit which records sync with Salesforce. If you sync only from Outlook to Salesforce, whenever users move a record out of the synced Outlook folder, that record is then deleted from Salesforce.

  If you set up the sync to be both ways or set the sync from Salesforce to Outlook whenever users delete a record in Salesforce, then that record is then deleted in Outlook.

If records are undeleted in Salesforce or they become available because of a change in sharing permissions, these records will appear in Outlook within 24 hours or when users manually sync using the icon shown in the system tray (described in detail further on).

> One recommendation is to only allow syncing of contact information from Salesforce to Outlook. This prevents population of users' personal contact information in Salesforce and prevents accidental deletion of key Salesforce contact records.

- **Setting of a common time zone for users in Salesforce and Outlook**

  Ensure that for every user that is going to sync using Salesforce for Outlook, Windows has a matching time zone set on their local computer to the time zone that is set for their user account in Salesforce.

  > If the Windows time zone does not match the time zone in the Salesforce CRM user profile, the syncing of events will fail.

- **Automatic insertion of e-mails into Salesforce when your users e-mail their contacts in Outlook**

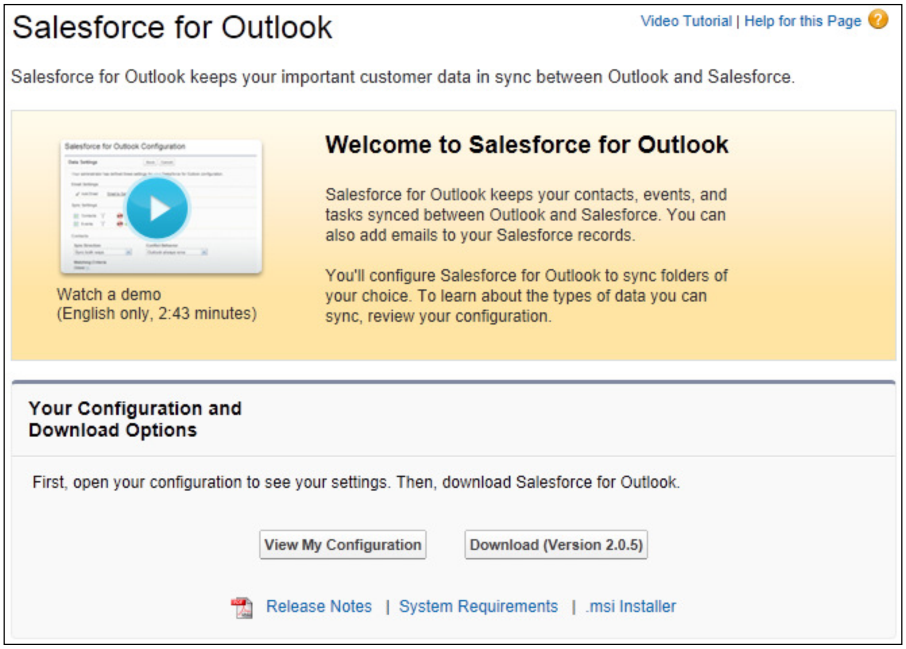
  You can set up the integration so that users' e-mails sent from Outlook are automatically inserted into Salesforce. The option presents an **Add to Salesforce** button from within Outlook and when activated the Salesforce for Outlook facility will attempt to associate e-mails to Salesforce contacts, leads, and/or opportunity records by checking the e-mail address that has been set using both the **To** and **Cc** fields within the e-mail. This is implemented using the Salesforce Email to Salesforce feature.

How to do it...

Carry out the following steps to install the Salesforce for Outlook software on a local machine:

1. Close Outlook.

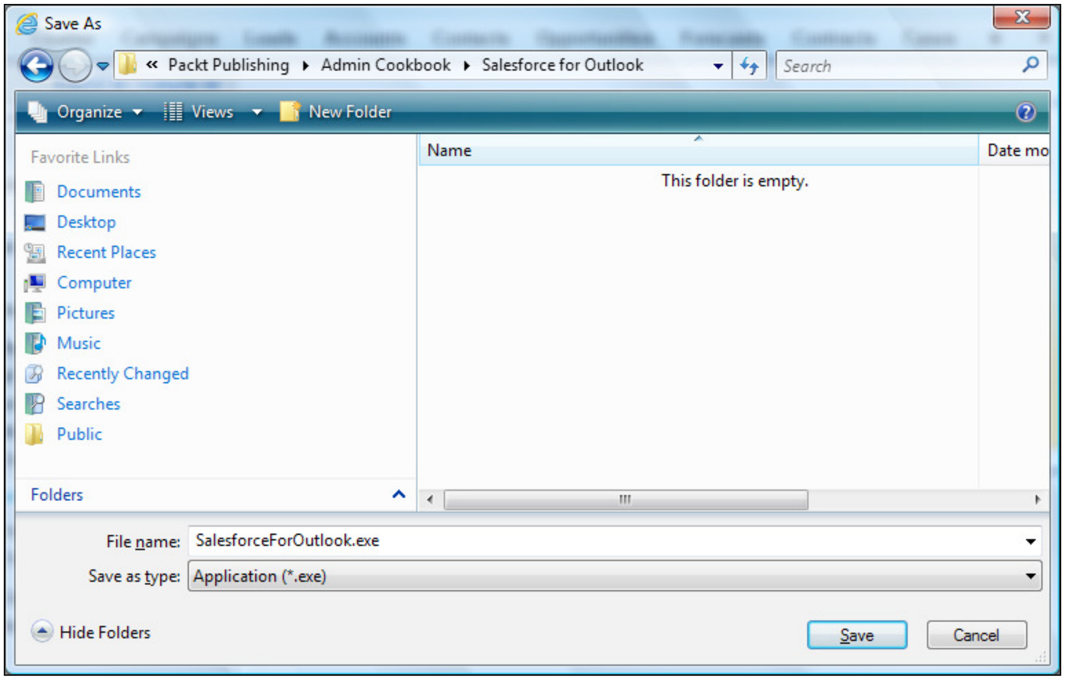
2. Navigate to the Salesforce for Outlook download page by going to **Your Name** | **Setup** | **Desktop Integration** | **Salesforce for Outlook**.



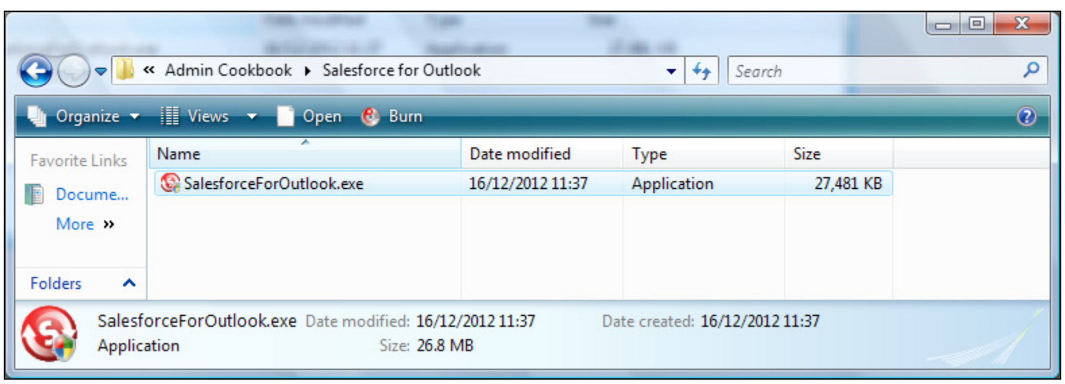
3. Click on **Download**.

> Here, we are proceeding to click the **Download (Version 2.0.5)** button.

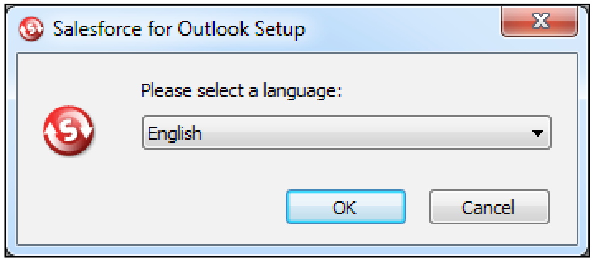
4. Depending on your browser type, choose the option to save the file and select a folder to save the the installation `.exe` file as shown in the following screenshot:



5. Click on **Save**.
6. Navigate to the folder and click on the **SalesforceForOutlook.exe** file as shown:



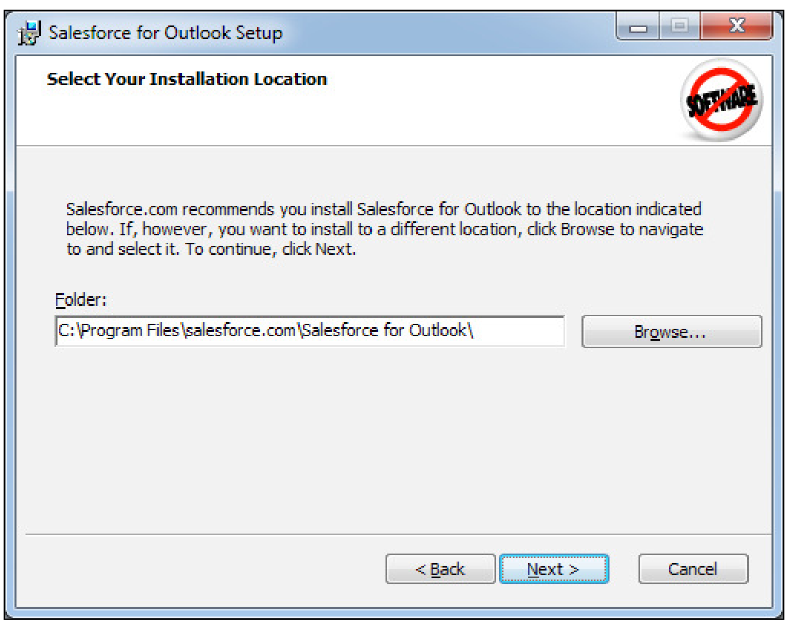
7. Select a language and click on **OK** as shown in the following screenshot:



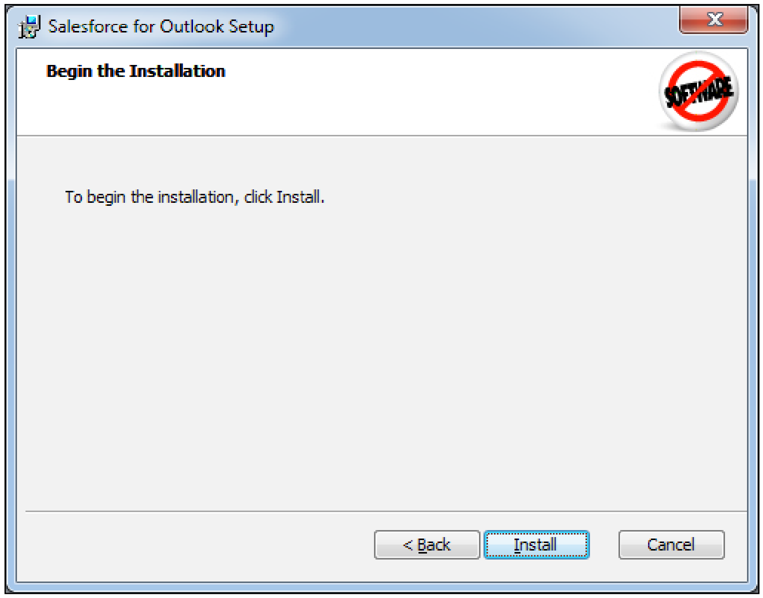
8. Click on **Next** when prompted as shown in the following screenshot:

9. Select the option to accept the terms in the license agreement and click on **Next** as shown in the following screenshot:

10. Provide the location for installation on the **Select Your Installation Location** page as shown in the following screenshot:



11. Click on **Next**.
12. Click on **Install** to complete the installation as shown in the following screenshot:



13. Finally, leave the **Start Salesforce for Outlook** checkbox checked and click on **Finish** as shown in the following screenshot:

14. After the setup wizard completes, a Salesforce for Outlook shortcut appears on the desktop with a red icon (shown on the left of the screenshot) and a small gray icon appears in the system tray (at the bottom-right of the screen as in the following screenshot):

Configuring the Salesforce for Outlook software on a local machine

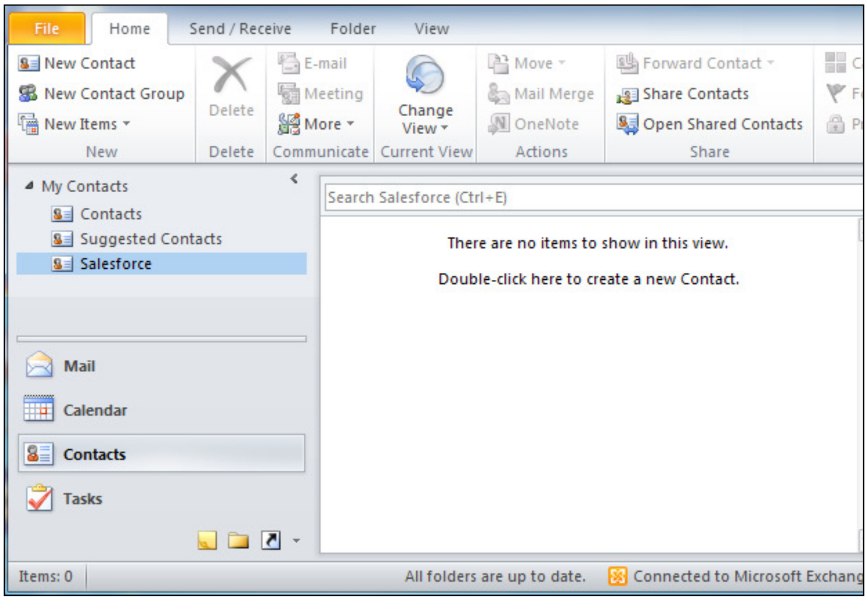
Now that we have installed the Salesforce for Outlook software, we can now proceed to configure the software on our local machine.

How to do it...

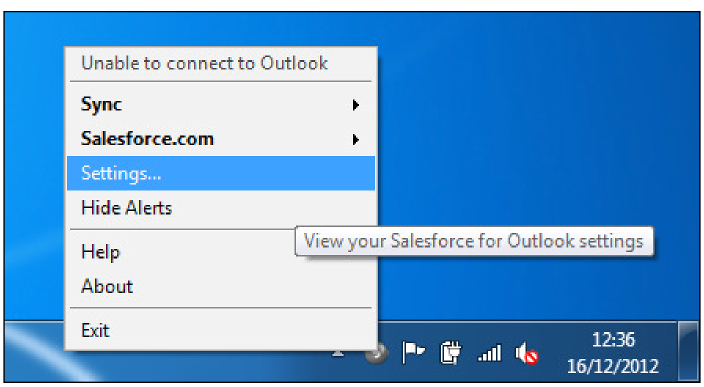
Carry out the following steps to configure the Salesforce for Outlook software on a local machine:

1. Open Outlook.

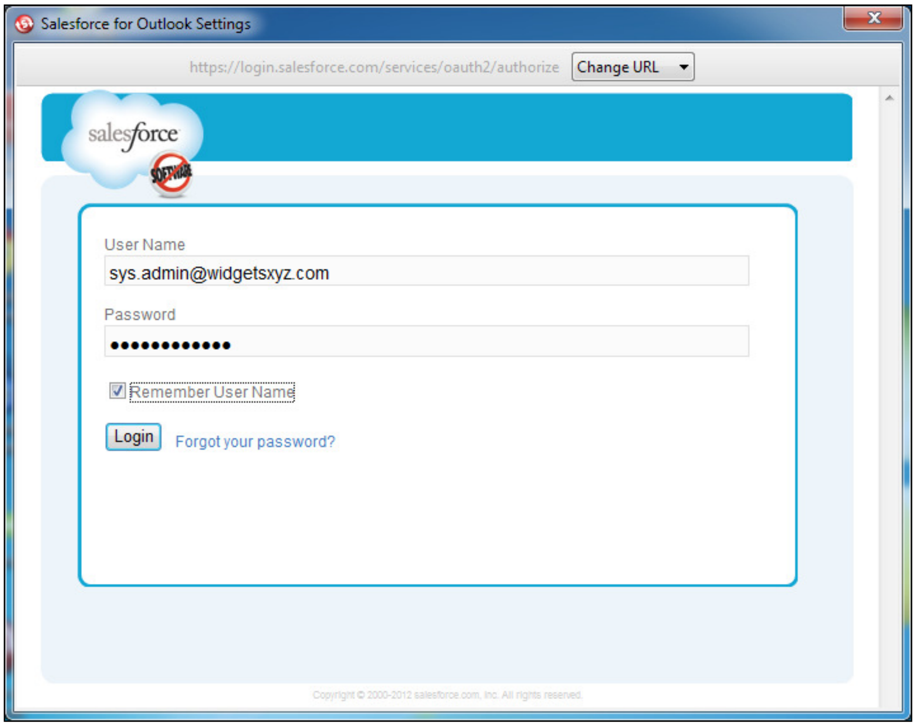
[Ensure that you have considered the creation of a Salesforce-specific contacts folder as described in the *Getting ready* section of the *Installing the Salesforce for Outlook software on a local machine* recipe. You can do this using the folder called **Salesforce** (for example) as shown in the following screenshot:]



2. Now, right-click on the small gray icon located in the system tray (as shown in step 14 of the previous recipe) and click on **Settings...** as shown in the following screenshot:

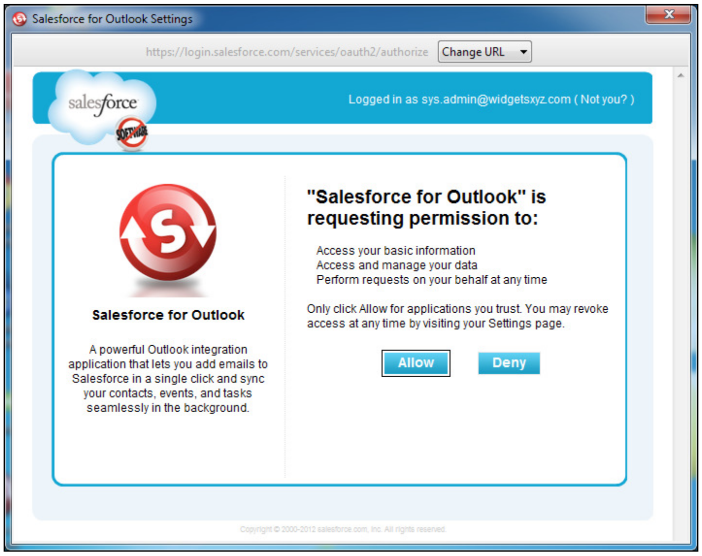


3. A login window appears in which you need to enter your Salesforce username and password. Optionally, check the **Remember User Name** checkbox as shown in the following screenshot:

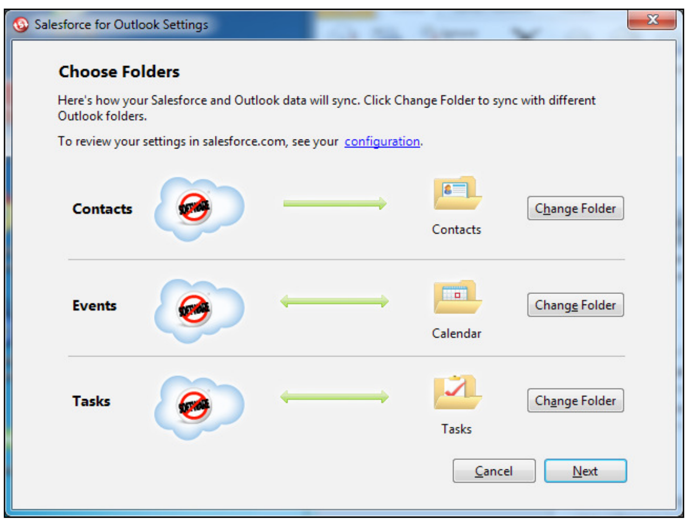


4. Click on **Login**.

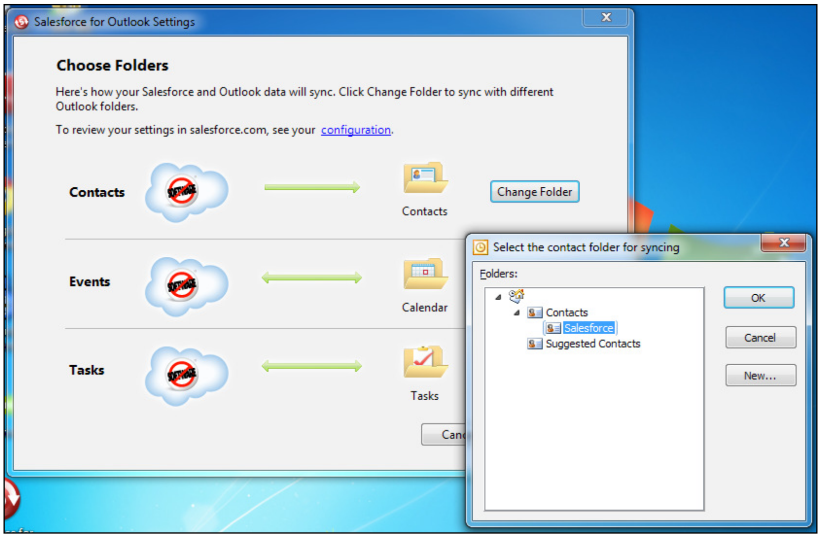
5. A screen stating **"Salesforce for Outlook" is requesting permission to:** is shown. This establishes a secure connection between Outlook and Salesforce, and once established and successfully connected, you will not have to log in again (if you click on **Deny**, you are returned to the previous screen):



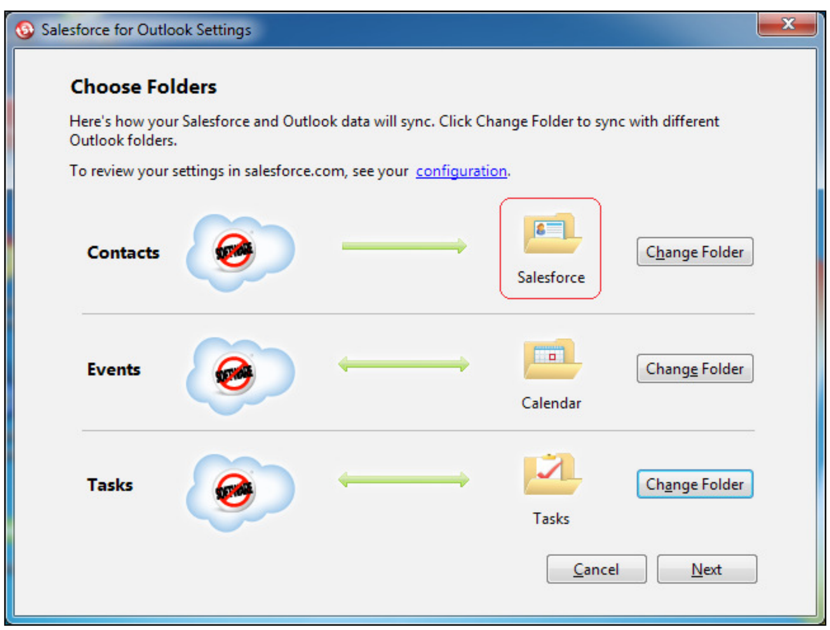
6. Click on **Allow**.
7. The configuration that was set up in Salesforce is now retrieved and a new screen is displayed to allow the setting up of the folders in Outlook that are to be synced as shown in the following screenshot:



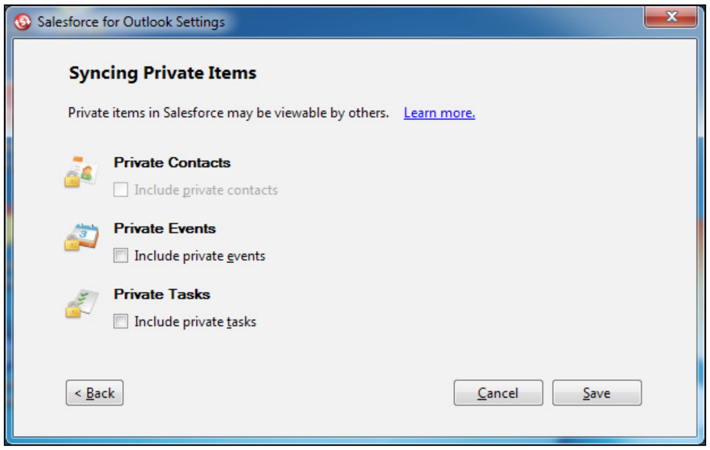
8. For contacts, click on **Change Folder**.
9. Select the **Salesforce** folder as shown in the following screenshot:



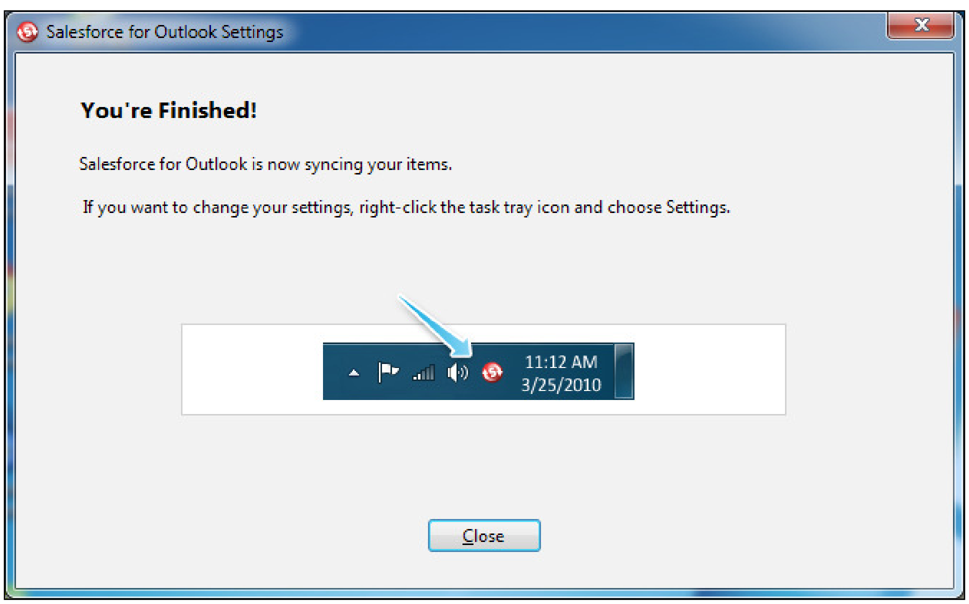
10. Click on **OK**.
11. Salesforce is then displayed as the contacts sync folder as shown in the following screenshot:



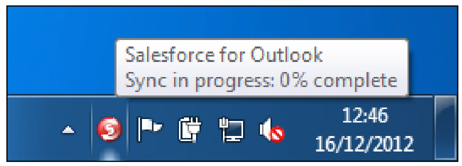
12. Click on **Next**.
13. Do not allow the syncing of private items. Here, do not check these checkboxes as shown in the following screenshot:



14. Finally, click on **Save**.
15. The finished screen is now displayed and the gray icon in the system tray changes color (to red) to show that it is now active as shown in the following screenshot:

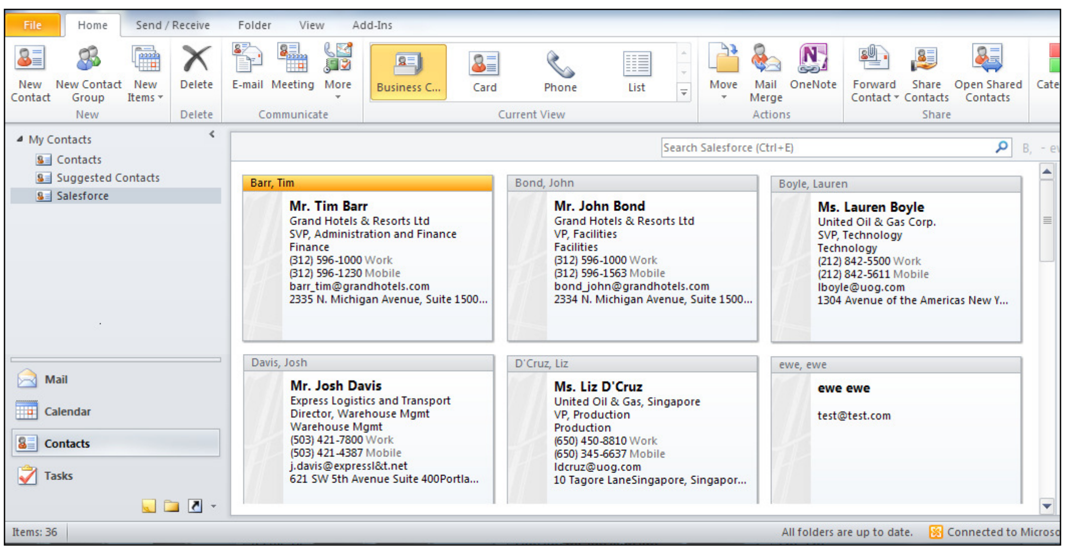


16. When Salesforce for Outlook is correctly installed and configured to sync Salesforce and Outlook items, the records in the selected folders automatically start to sync.

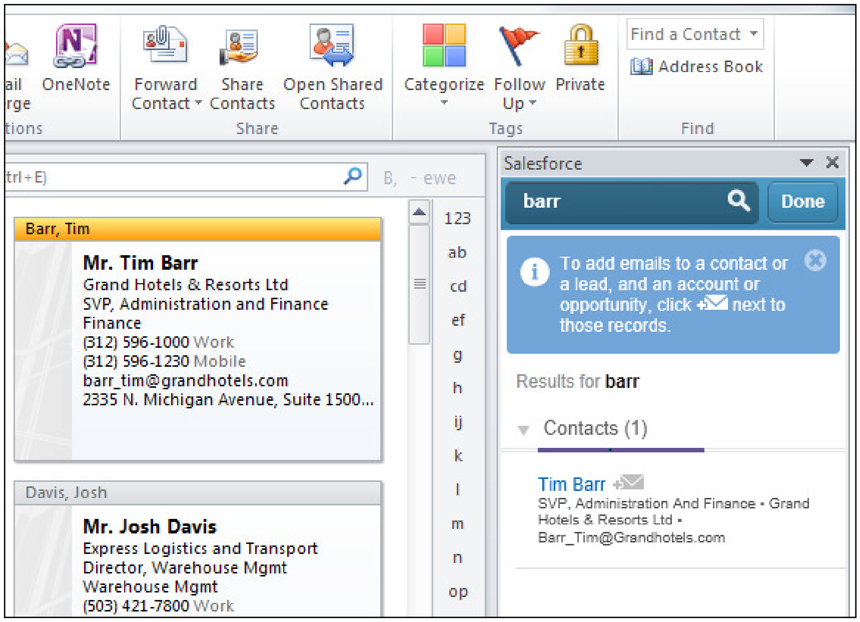
> The Salesforce for Outlook icon in the system tray spins to indicate that records are being synced and a hover pop up message showing the percentage complete is displayed as shown in the following screenshot:
>
> 
>

How it works...

When the sync is complete, the contact records in Salesforce appear in the **Salesforce** folder that we created previously as shown in the following screenshot:

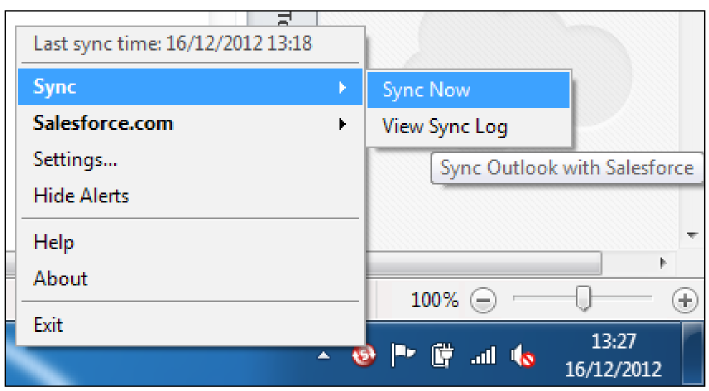


We can search for Salesforce contacts within Outlook using the **Side-Panel (Beta)** feature as shown in the following screenshot:

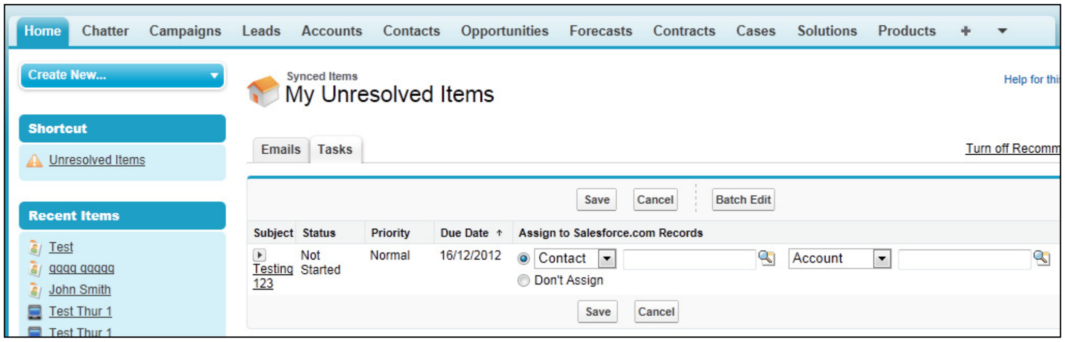


There's more...

You can force a manual sync at any time by right-clicking on the Salesforce system tray icon and clicking on **Sync** | **Sync Now** as shown in the following screenshot:



Tasks and events are not automatically associated to related Salesforce records when the sync between Outlook and Salesforce takes place. Users have the option instead to manually assign or set as not assigned using the **My Unresolved** items menu (which is accessed from the left-hand side bar as shown in the following screenshot):



If data is undeleted in Salesforce or becomes available because of a change in sharing permissions, it will show up in Outlook within 24 hours or when you manually sync from the system tray.

9

Integrating Salesforce CRM with External Online Tools

In this chapter, we will cover the following recipes:

- Providing a Google News search for an account using a custom link
- Building a custom Web Tab to display an external web application
- Displaying the location of an organization using a Google Map and a Visualforce page

Introduction

In this chapter, we provide recipes that integrate Salesforce CRM with external online tools which extend the power of the CRM platform and provide enhanced functionality for your users.

These recipes provide features and functionality that present information stored outside Salesforce using mechanisms that are not provided natively within Salesforce.

Providing a Google News search for an account using a custom link

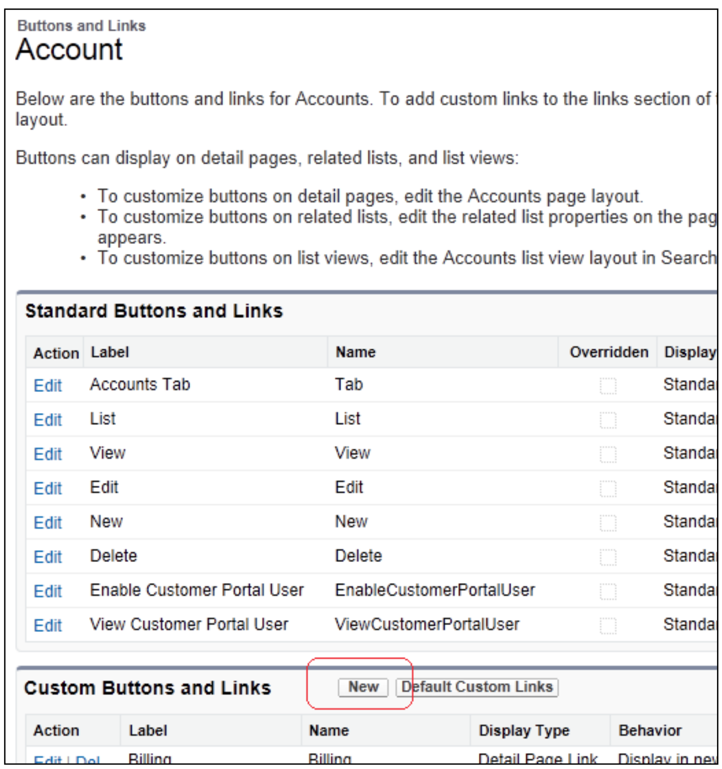
When dealing with customers it is always useful for the users of Salesforce CRM to be aware of the latest news related to the customer. News information related to customers can be retrieved from sources such as Google News and is useful for all users whether from the sales team, marketing, or perhaps the finance team.

In this recipe we will provide a link, displayed on an **Account Detail** page, that when clicked opens a new window showing a Google News result for the name of the account.

How to do it...

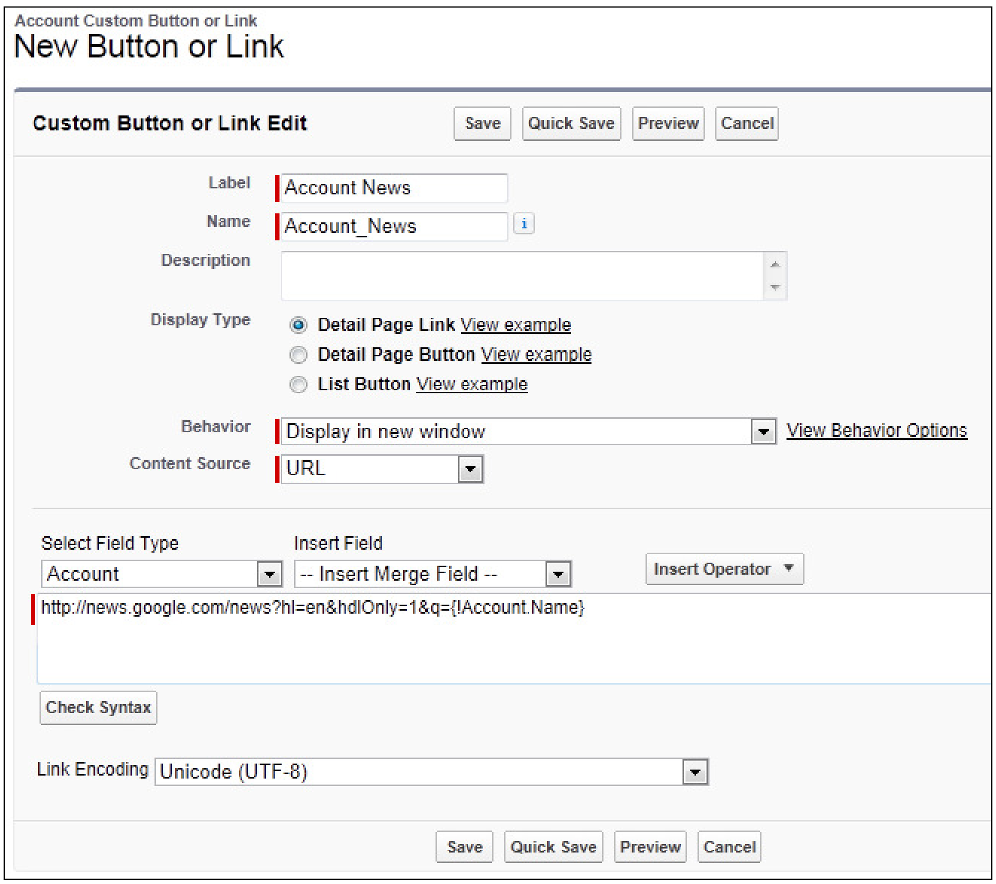
Carry out the following steps to create a Google News search for an account using a custom link:

1. Navigate to the custom web tab setup page by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Buttons and Links**. Scroll down to the **Custom Buttons and Links** section as shown in the following screenshot:

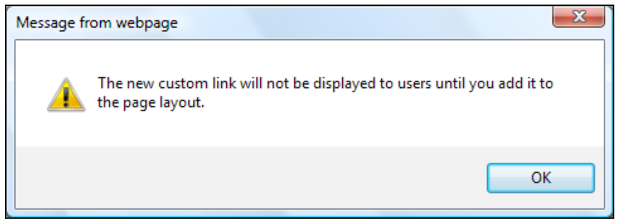


2. Click on **New**, as shown in the preceding screenshot.
3. Enter `Account News` in the **Label** field.
4. Accept the default **Account_News** in the **Name** field.
5. Choose the option of **Detail Page Link** in the **Display Type** options choices.
6. Select the value **Display in new window** in the **Behavior** picklist.
7. Select the value **URL** in the **Content Source** picklist.
8. In the URL textbox, paste the following code (as shown in the screenshot further below):

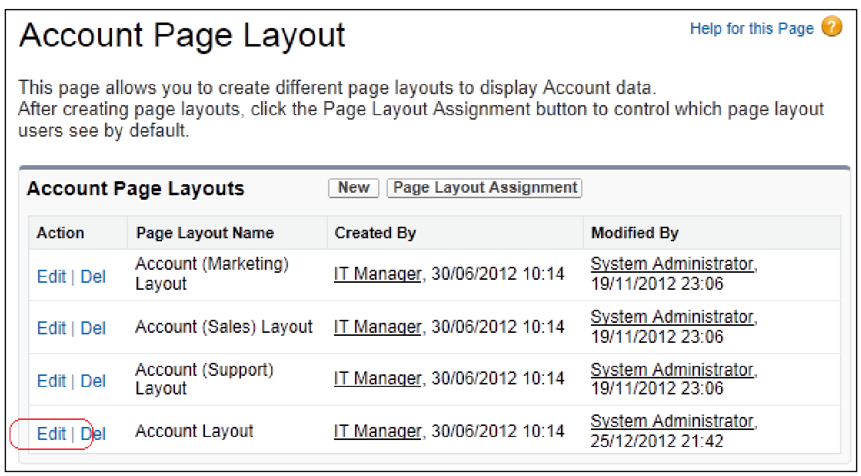
```
http://news.google.com/news?hl=en&hdlOnly=1&q={!Account.Name}
```



9. Click on **Save**.

> Upon saving, we are reminded that the custom link will not be displayed to users until you add it to the page layout (as shown in the following screenshot).
>
> 
>

10. Click on **OK**.
11. We can now add the custom link to the page layout within the account page layout setup page by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Page Layouts**.
12. Determine which Account Page Layout to include the Google News search custom link on and click on **Edit**.

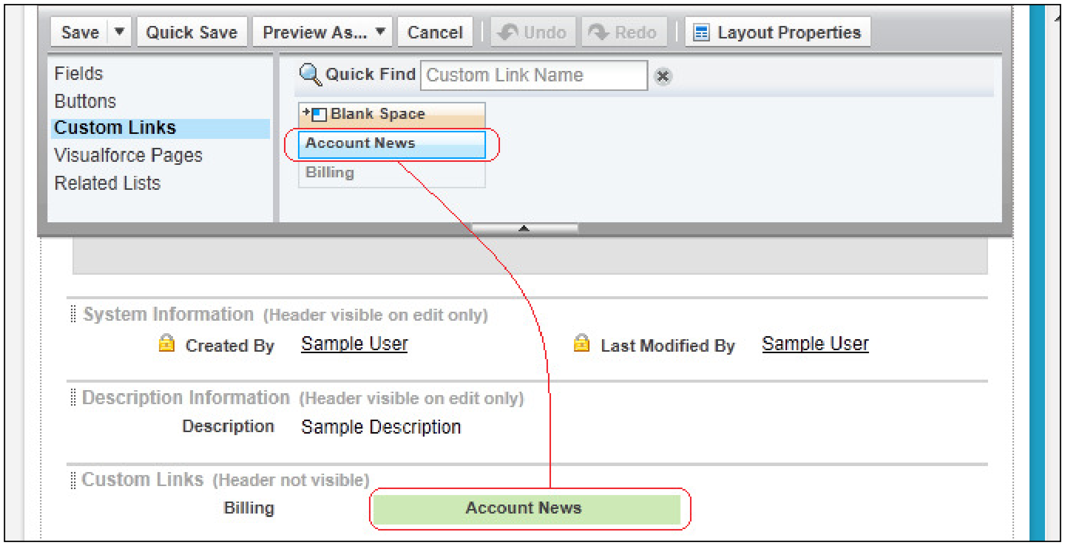


Here we are editing the home page layout named Account Layout (as shown in the preceding screenshot).

13. Click on **Edit**.

We are now presented with the Account Layout edit page.

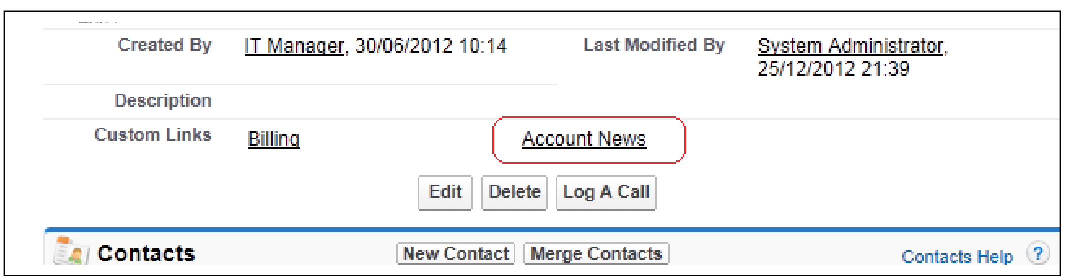
14. Click on the **Custom Links** text at the top-left section of the page as shown in the following screenshot:



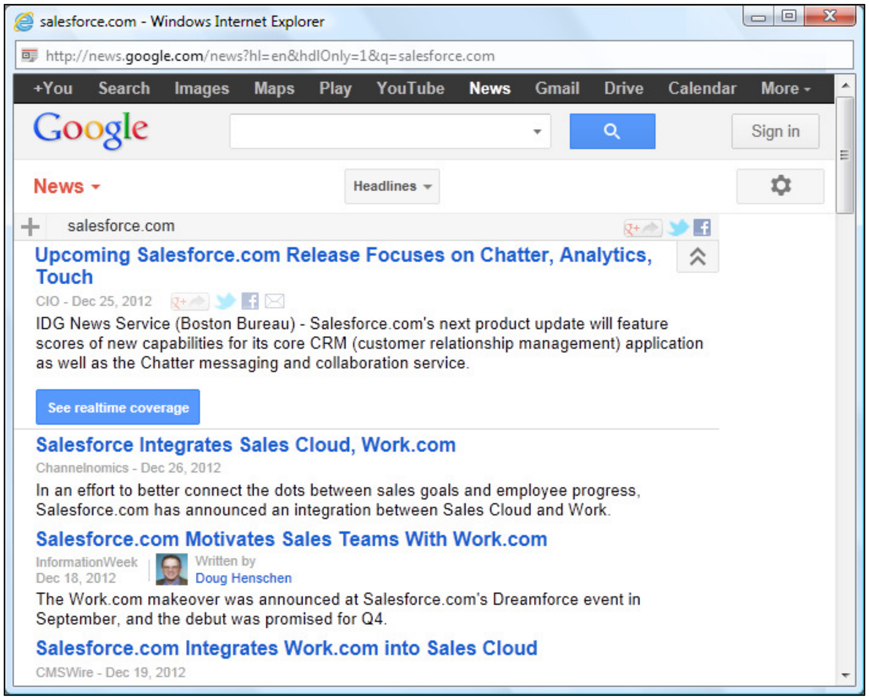
15. Locate the **Account News** custom link and drag it to the **Custom Links** section (as shown in the preceding screenshot).
16. Click on **Save**.

How it works...

The **Account News** custom link now appears in the **Custom Links** area of the main **Account Detail** page (before the related lists) for each Account record, as shown in the following screenshot:



When clicking the link, the name of the account is passed to the Google News search page which is then opened in a new window as shown in the following screenshot:



Building a custom Web Tab to display an external web application

Having a custom Web Tab enables the display of web content or applications embedded within the Salesforce CRM application window.

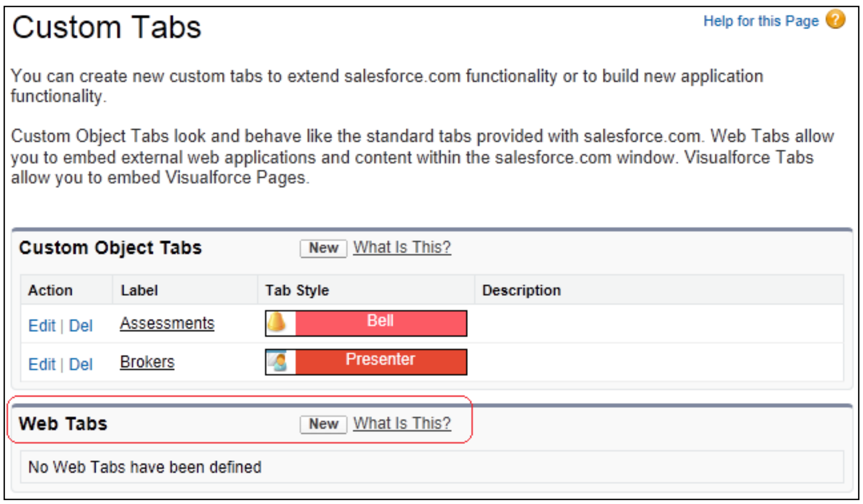
You can create custom Web Tabs to allow Salesforce users to view the company intranet or a wiki that is used frequently so that they can find the information they need without leaving the Salesforce CRM application.

In this recipe we will display an example Web Tab where we pass information specified from within Salesforce, the purpose of which is to demonstrate the mechanism involved in setting up custom Web Tabs for your specific external web applications.

How to do it...

Carry out the following steps to create a Web Tab to display an external web application:

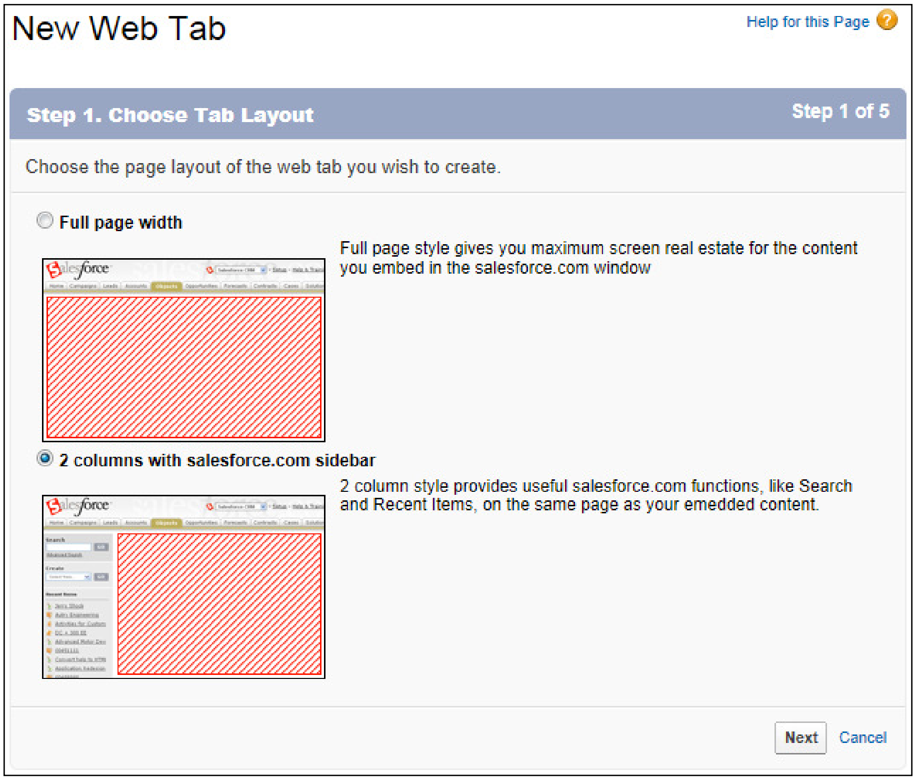
1. Navigate to the custom web tab setup page by going to **Your Name** | **Setup** | **Create** | **Tabs**.



2. Scroll down to the **Web Tabs** section as shown in the preceding screenshot.
3. Click on **New**.

> We will be presented with the **Step 1. Choose Tab Layout** page. Here you can either choose to have the web page content run across the **Full page width**, or as **2 columns with salesforce.com sidebar** to help users navigate within Salesforce CRM (as shown in the screenshot further below).

4. Select the **2 columns with salesforce.com sidebar** option (as shown in the following screenshot).

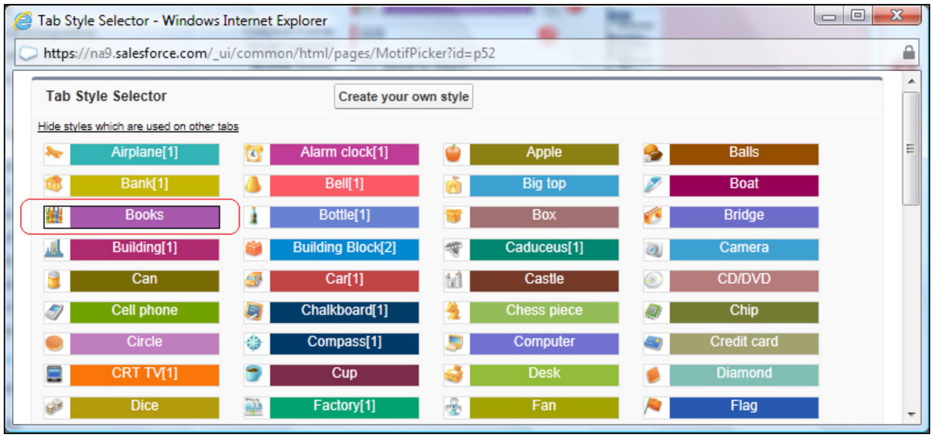


5. Click on **Next**.

> We will be presented with the **Step 2. Define Content and Display Properties** page (as shown in the screenshot further below).

6. Enter **CRM Books** in the **Tab Label** textbox.
7. Accept the default **CRM_Books**, auto populated text in the **Tab Name** text box.

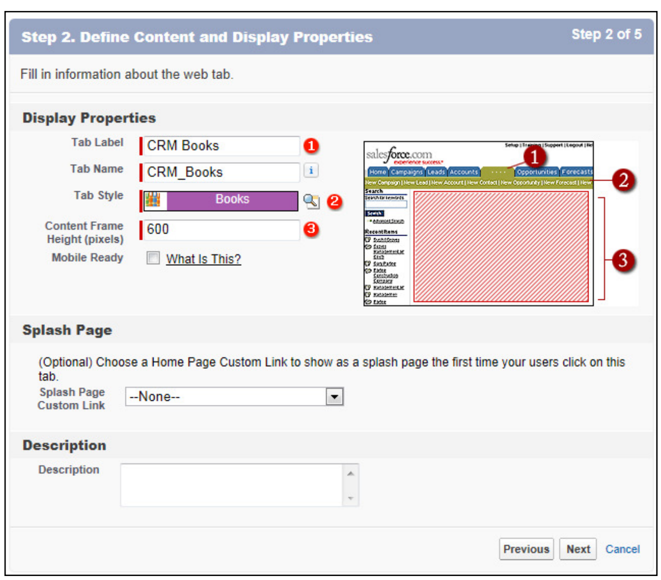
8. In the **Tab Style** field, click on the lookup icon and select an icon for the new Web Tab. Here we are selecting the **Books** style (as shown in the following screenshot).



9. Leave the **Content Frame Height (pixels)** setting at **600**.

> You can make Web Tabs available in the mobile application by selecting the **Mobile Ready** checkbox. However, for this recipe, we are not making this Web Tab mobile ready.

10. Optionally select a splash page and add a description. Here we are leaving these sections blank (as shown in the following screenshot).

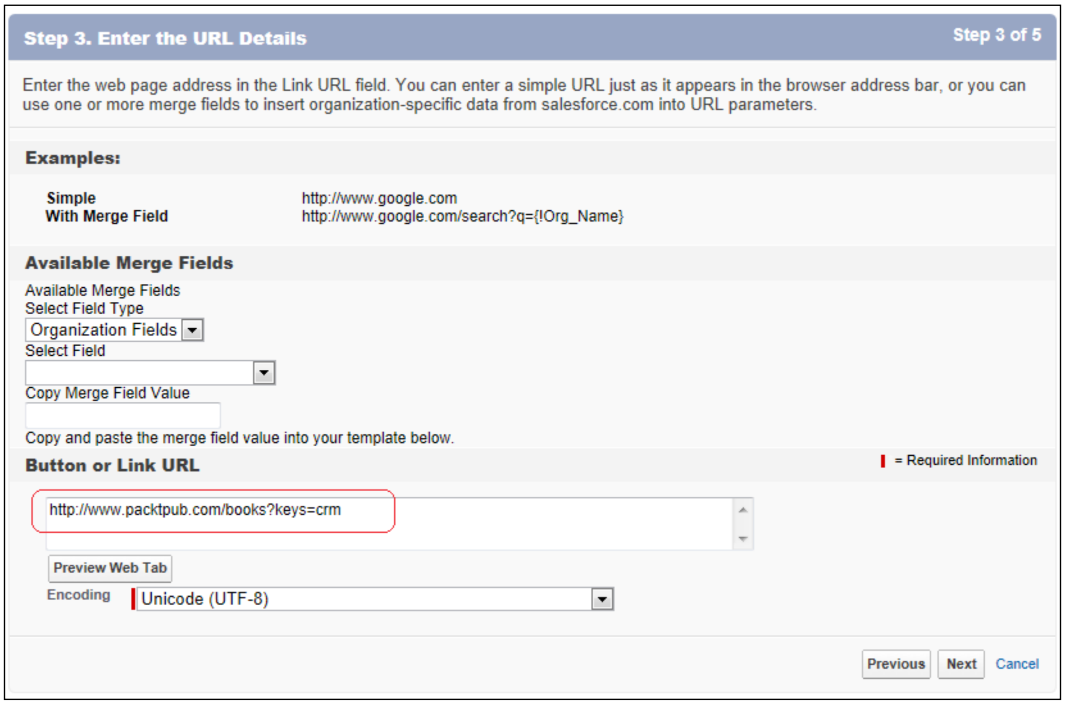


11. Click on **Next**.

> We will be presented with the **Step 3. Enter the URL Details** page (as shown in the screenshot further below).

12. Paste the following web URL into the **Button or Link URL** textbox (as shown in the following screenshot).

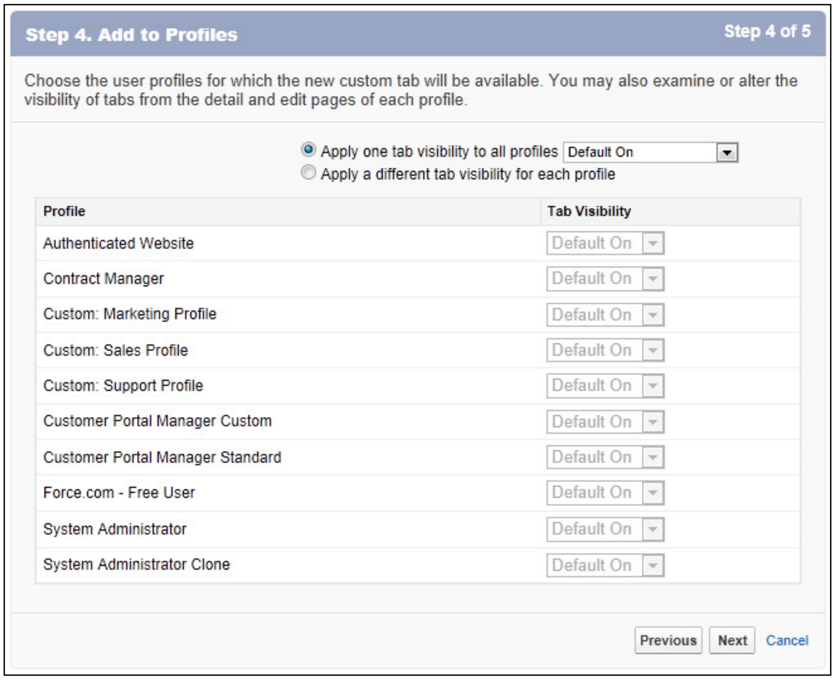
    `http://www.packtpub.com/books?keys=crm`



13. Click on **Next**.

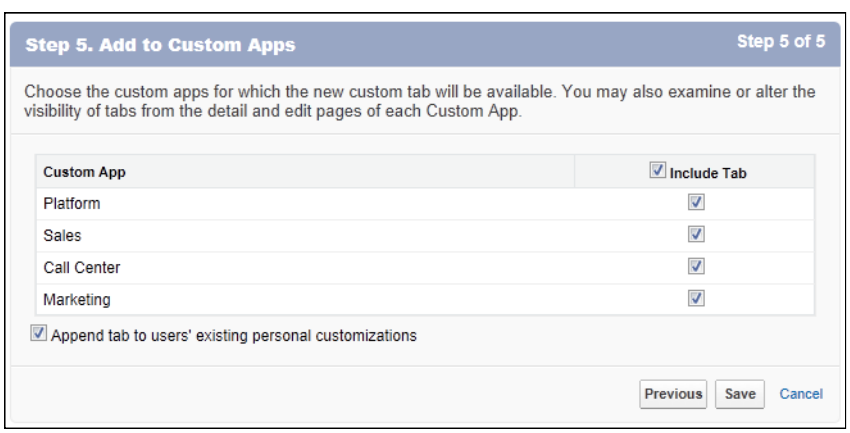
> We will be presented with the **Step 4. Add to Profiles** page (as shown in the screenshot further below).

14. Select the profiles to which you want to grant visibility to this web tab. Here, we are setting tab visibility for all profiles (as shown in the following screenshot).



15. Click on **Next**.

We will be presented with the **Step 5. Add to Custom Apps** page (as shown in the following screenshot).



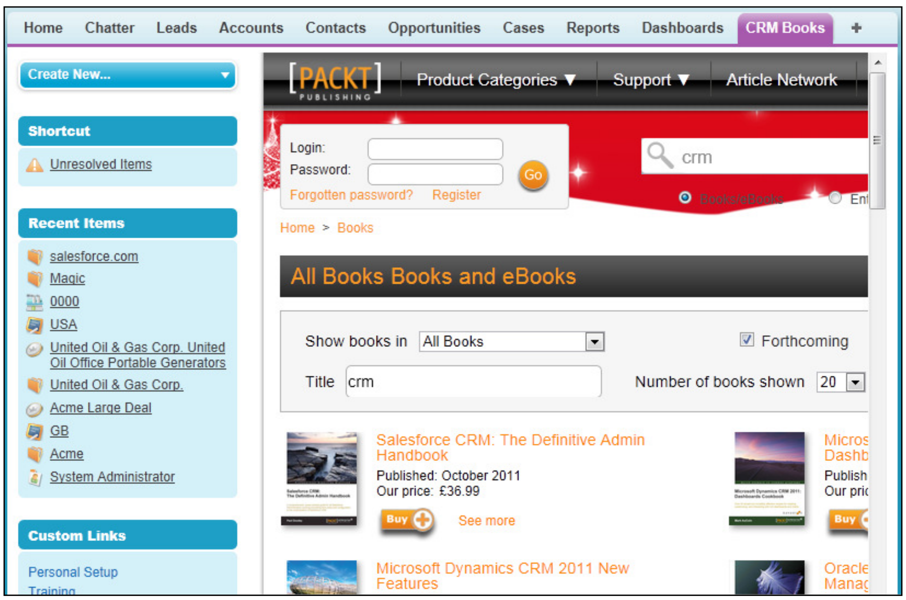
Select only the Apps that you wish to be included on the Custom Web Tab.

16. Finally, click on **Save**.

How it works...

The custom Web Tab **CRM Books** now appears in the list of tabs and upon clicking this tab, the URL that is specified is accessed, with the web page content retrieved into Salesforce CRM.

You can see how this looks in the following screenshot:



Displaying the location of an organization using a Google Map and a Visualforce page

When dealing with customers it is always useful for the users of Salesforce CRM to be aware of the location of the customer. Address information related to customers can be used to retrieve map details from sources such as Google and is useful for all users whether from the sales team, marketing, or perhaps the finance team.

In this recipe we will provide a Visualforce page, displayed on an **Account Detail** page, that displays a Google Map showing a location marker for the address of the account.

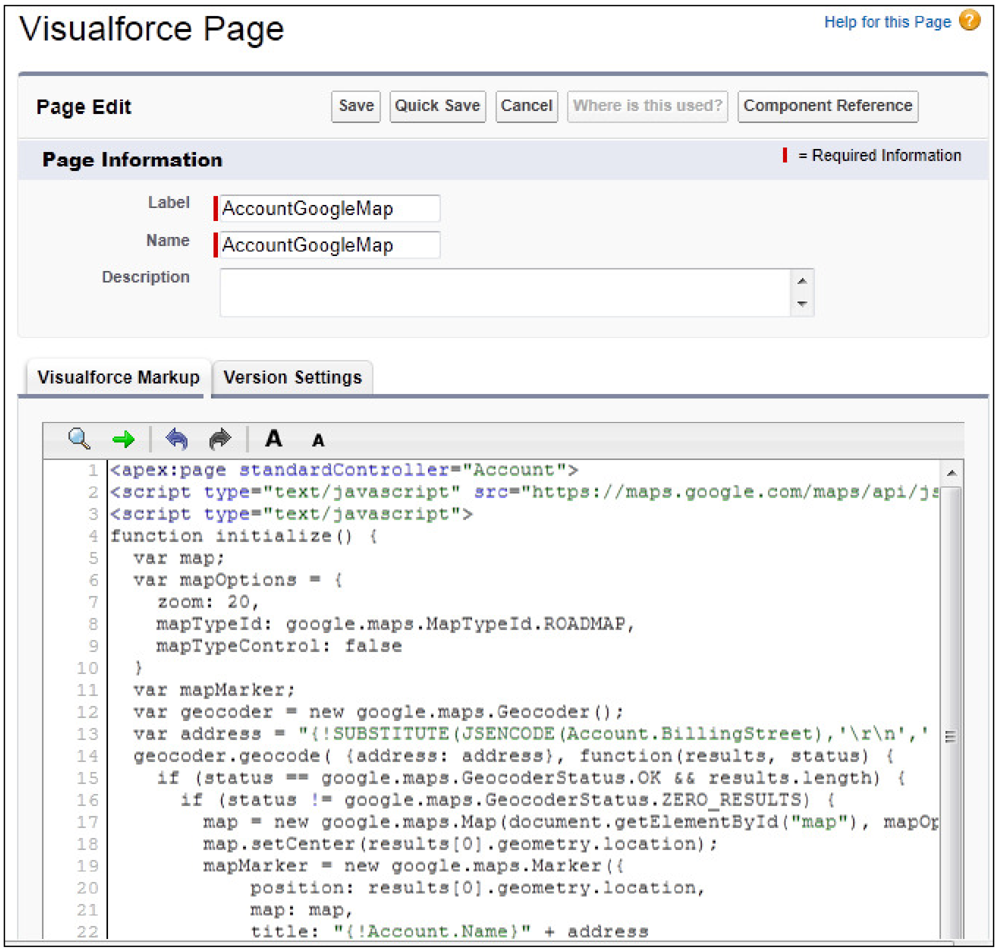
How to do it...

Carry out the following steps to display the location of an organization using a Google Map and a Visualforce page:

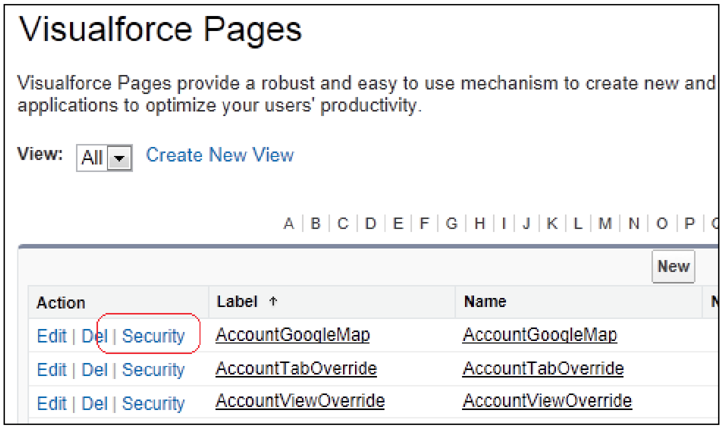
1. Navigate to the Visualforce Pages setup screen by going to **Your Name** | **Setup** | **Develop** | **Pages**.
2. Click on **New**.
3. Enter `AccountGoogleMap` in the **Label** field.
4. Accept the default **AccountGoogleMap** in the **Name** field.
5. Paste the following code (as shown in the following screenshot):

```
<apex:page standardController="Account">
<script type="text/javascript" src="https://maps.google.com/maps/
api/js?sensor=false"></script>
<script type="text/javascript">
function initialize() {
  var map;
  var mapOptions = {
    zoom: 13,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    mapTypeControl: false
  }
  var mapMarker;
  var geocoder = new google.maps.Geocoder();
  var address = "{!SUBSTITUTE(JSENCODE(Account.BillingStreet),'\
r\n',' ')}, " + "{!Account.BillingCity}, " + "{!
Account.BillingPostalCode}, " + "{!Account.BillingCountry}";
  geocoder.geocode( {address: address}, function(results, status)
{
    if (status == google.maps.GeocoderStatus.OK && results.length)
{
      if (status != google.maps.GeocoderStatus.ZERO_RESULTS) {
        map = new google.maps.Map(document.getElementById("map"),
mapOptions);
        map.setCenter(results[0].geometry.location);
        mapMarker = new google.maps.Marker({
            position: results[0].geometry.location,
            map: map,
            title: "{!Account.Name} " + address
        });
      }
    } else
```

```
        document.getElementById("map").innerHTML = "Unable to find
or display a map for {!Account.Name}'s billing address : " +
address;
  });
}
</script>
<div id="map" style="width:100%;height:300px"></div>
<script>
    initialize();
</script>
</apex:page>
```



6. Click on **Save**.
7. Now set the security for the required profiles in your organization by going to **Your Name** | **Setup** | **Develop** | **Pages**.
8. Locate the row for the Visualforce Page **AccountTabOverride** and click on the **Security** link, as shown in the following screenshot:



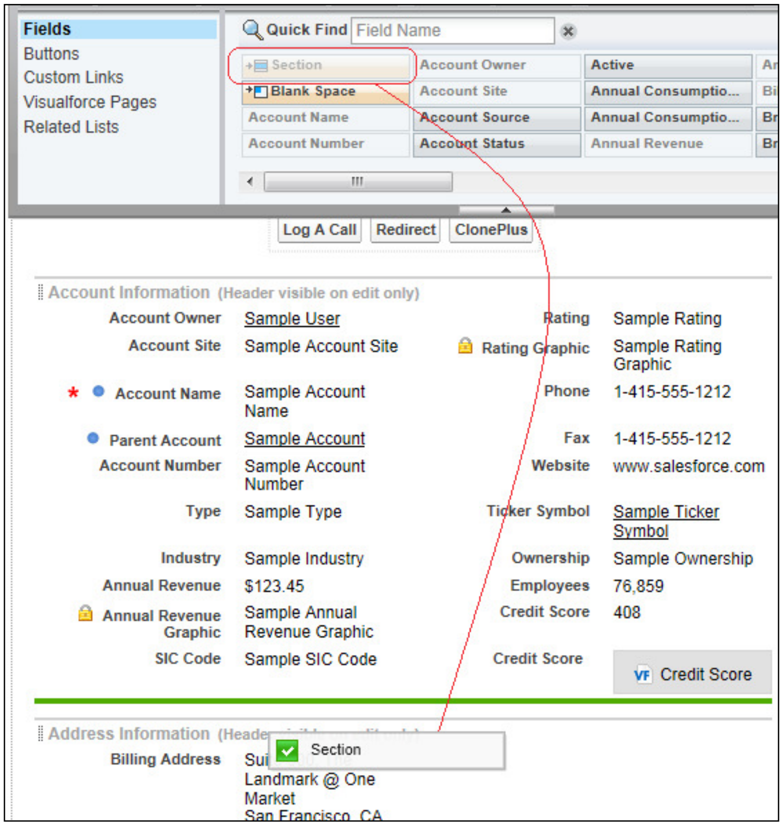
9. Set the security for the required profiles.
10. Click on **Save**.

To display the Account Google Map within an account record we'll add the page to a new section on an Account Page Layout by carrying out the following steps:

1. Navigate to the Account Page Layout setup page by going to **Your Name** | **Setup** | **Customize** | **Accounts** | **Page Layouts**.
2. Determine which Account Page Layout to place the Account Google Map Visualforce page on and click on **Edit**.

> Here we are editing the home page layout named **Account Layout**.

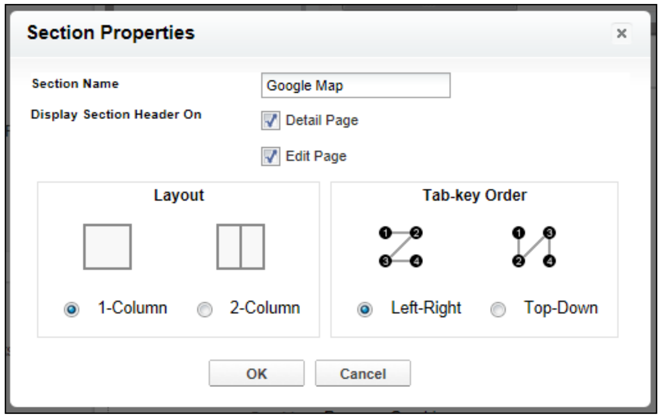
3. Create a new section by dragging a **Section** icon from the top menu area into an appropriate position on the Account Detail page, as shown in the following screenshot:



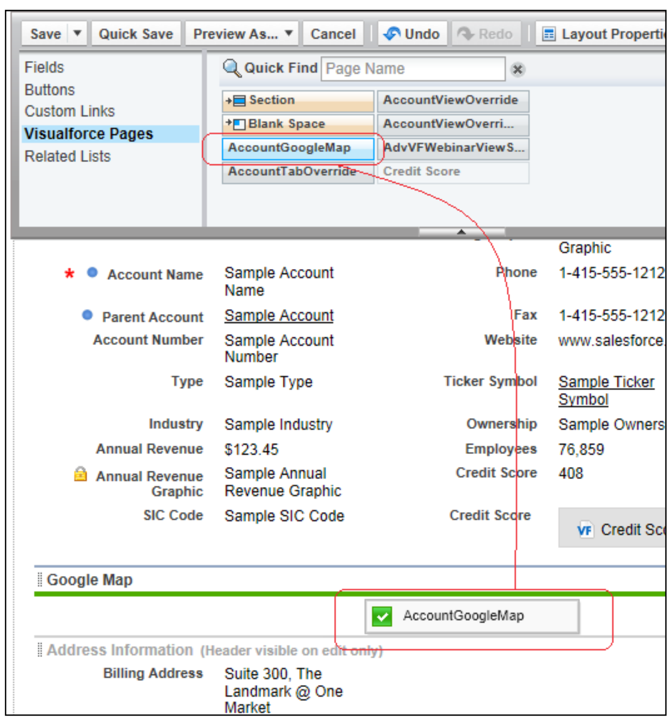
[Upon placing the new section we will be presented with the **Section Properties** page (as shown in the following screenshot).]

4. Enter **Google Map** in the **Section Name** field.

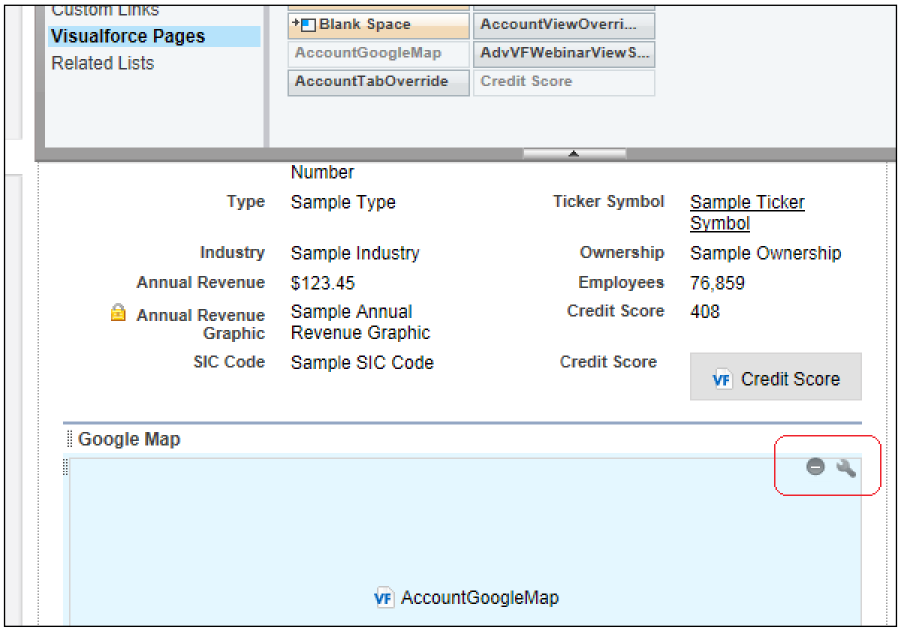
5. Choose the **Layout** option as **1-Column** (as shown in the following screenshot):



6. Click on **OK**.
7. Click on the **Visualforce Pages** text at the top-left section of the page as shown in the screenshot further below.
8. Locate the **AccountGoogleMap** Visualforce page and drag it to the **Google Map** section (created previously) as shown in the following screenshot:



> Upon placing the Visualforce page, we will be presented with the following screen, where we can change the properties of the Visualforce page.



9. Locate the main buttons at the top of the screen.
10. Click on **Save**.

How it works...

The lines of code that display the location of an organization using a Google Map and a Visualforce page can be seen in the following screenshot:

```
<apex:page standardController="Account">
<script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
function initialize() {
  var map;
  var mapOptions = {
    zoom: 13,
    mapTypeId: google.maps.MapTypeId.ROADMAP,
    mapTypeControl: false
  }
  var mapMarker;
  var geocoder = new google.maps.Geocoder();
  var address = "{!SUBSTITUTE(JSENCODE(Account.BillingStreet),'\r\n',' ')}, " +
  "{!Account.BillingCity}, " + "{!Account.BillingPostalCode}, " + "{!Account.BillingCountry}";
  geocoder.geocode( {address: address}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK && results.length) {
      if (status != google.maps.GeocoderStatus.ZERO_RESULTS) {
        map = new google.maps.Map(document.getElementById("map"), mapOptions);
        map.setCenter(results[0].geometry.location);
        mapMarker = new google.maps.Marker({
            position: results[0].geometry.location,
            map: map,
            title: "{!Account.Name} " + address
        });
      }
    } else
        document.getElementById("map").innerHTML = "Unable to find or display a map for
        {!Account.Name}'s billing address : " + address;
  });
}
</script>
<div id="map" style="width:100%;height:300px"></div>
<script>
    initialize();
</script>
</apex:page>
```

The following section describes how this set of code works.

```
<apex:page standardController="Account">
```

This is the opening tag for our Visualforce page which uses a standard Controller for the Account Object. This allows the page to be embedded into the **Account Details** page and access the values from the record. For example, `{!Account.BillingCity}`.

```
<script type="text/javascript" src="https://maps.google.com/maps/api/
js?sensor=false"></script>
```

This calls the JavaScript library that allows the Google Map functionality to appear on our web page:

```
<script type="text/javascript">
function initialize() {
```

This is the opening tag for the JavaScript function that our Visualforce page requires to call the Google Map code:

```
var map;
var mapOptions = {
  zoom: 13,
  mapTypeId: google.maps.MapTypeId.ROADMAP,
  mapTypeControl: false
}
```

The preceding code sets the options for the Google Map code where:

- `zoom`: This is used to specify the initial zoom level of the map. This value has to be an integer between `0` and `19` where `0` is fully zoomed out and `19` is fully zoomed in.
- `mapTypeId`: This is used to specify the type Google Maps. This value defines what the initial map type would be where the options are `ROADMAP` (which is a normal map); `SATELLITE` (which is a satellite image); `HYBRID` (which displays satellite images with roads and labels overlayed on it); `TERRAIN` (which displays a map with physical features such as terrain and vegetation).
- `mapTypeControl`: This is used to specify whether the mapTypeControl will be displayed or not. The `mapTypeControl` is the control positioned in the upper-right corner of the map from which you can choose what map type to show. Set it to `true` to display it and to `false` to hide it. Note that the default value is true.

  ```
  var mapMarker;
  var geocoder = new google.maps.Geocoder();
  ```

  This invokes the Google Map Geocoder object. To specify a location, Google Maps requires a latitude and longitude. So, our main code step will be to convert the physical Account Billing address into the necessary mapping format. This process has its own scientific term which is geocoding.

  ```
  var address = "{!SUBSTITUTE(JSENCODE(Account.BillingStreet),'\
  r\n',' ')}, " + "{!Account.BillingCity}, " + "{!Account.
  BillingPostalCode}, " + "{!Account.BillingCountry}";
  ```

This formats the Salesforce Account Billing address into a format suitable for passing to the Google Map JavaScript library. It uses the merge fields such as `{!Account.BillingCity}` to obtain the address data elements.

This first part which formats the Account Billing Street using the following `{!SUBSTITUTE(JSENCODE(Account.BillingStreet),'\r\n',' ')}` uses Salesforce functions to strip the multiline address of non JavaScript friendly characters and then replaces the newline characters with spaces. JSENCODE encodes text and merge field values for use in JavaScript by inserting escape characters, such as a backslash (`\`), before unsafe JavaScript characters, such as the apostrophe (`'`). `SUBSTITUTE` substitutes new text for old text in a text string.

```
geocoder.geocode( {address: address}, function(results, status) {
```

This is where the actual call for the geocoding starts. The `geocode` method uses an object `address` as well as a callback function to give the results of the mapping attempt. The parameter, `{ address : address }`, simply passes the address which we formulated above passed to the `address` object.

```
        if (status == google.maps.GeocoderStatus.OK && results.length) {
          if (status != google.maps.GeocoderStatus.ZERO_RESULTS) {
```

This line handles the results of the geocoding call. After the geocoding finishes, the callback function is invoked with a results array and status that is used to determine success or failure.

```
            map = new google.maps.Map(document.getElementById("map"),
    mapOptions);
```

Having reached this step of the code would signify a successful geocode. Using the `mapOptions` that we specified earlier, this line is where the map is constructed. The constructed map is passed to an HTML `div` tag which is covered further.

```
            map.setCenter(results[0].geometry.location);
            mapMarker = new google.maps.Marker({
                position: results[0].geometry.location,
                map: map,
                title: "{!Account.Name} " + address
            });
          }
        } else
```

The line of code above sets the marker for the map and uses the Salesforce Account Name merge field to set the title for the marker on the map.

```
            document.getElementById("map").innerHTML = "Unable to find or
    display a map for {!Account.Name}'s billing address : " + address;
```

This line only appears where the geocoding call failed to return a successful response and set the error message in the HTML `div` tag which is covered further below.

```
      });
    }
    </script>
    <div id="map" style="width:100%;height:300px"></div>
```

This is the HTML `div` tag called `map` which is where either the constructed Google Map code or our constructed error message is rendered. Here we have set the width to be the maximum width of the Salesforce detail page and a height of `300px`.

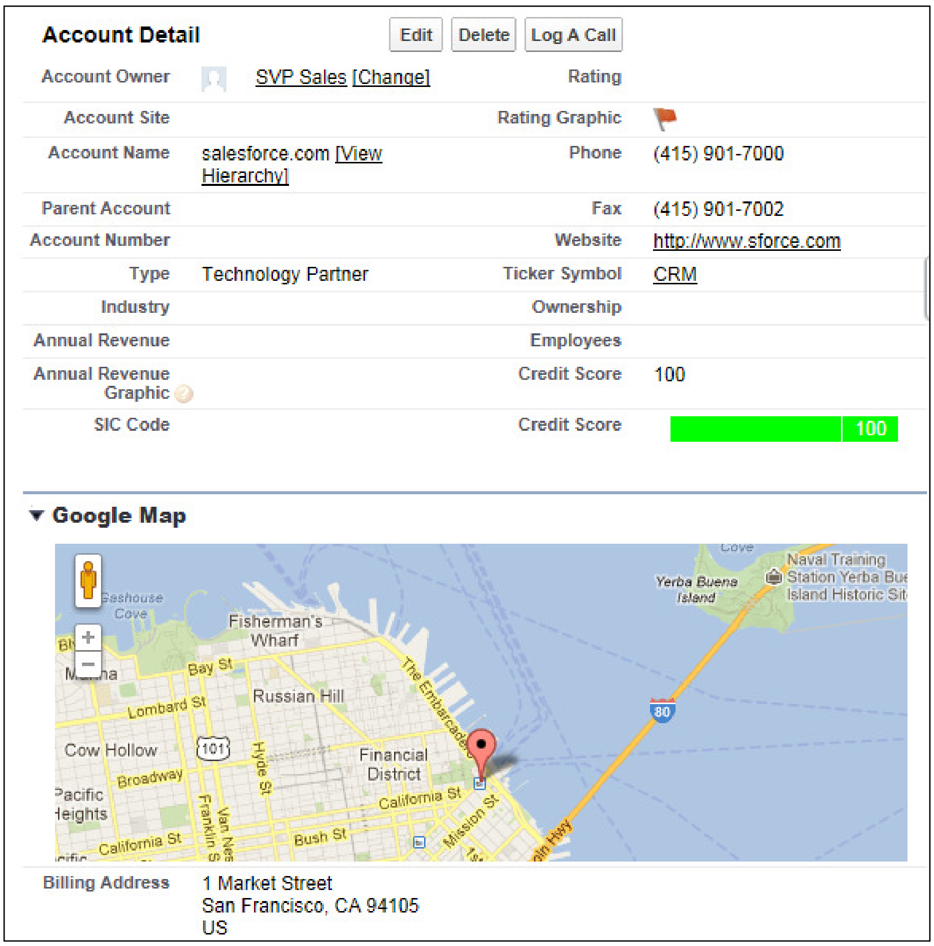
```
    <script>
        initialize();
```

This is the where our Visualforce page immediately invokes the calls to the Google Map JavaScript libraries and the function calls as specified above.

```
</script>
</apex:page>
```

This is the closing tag for our Visualforce page which uses a standard Controller for the Account Object.

The Account Google Map now appears on an Account Detail page (showing the address that is set in the Billing Address) for each Account record as shown in the following screenshot:



Index

Symbols

2 columns with salesforce.com sidebar option 230
15 digit Salesforce opportunity ID
converting to 18 digits, formula field used 174-176
18 digits
conversion from 15 digit Salesforce opportunity ID, formula field used 174-176
100k Deal Review template 163

A

account
Google News search providing, custom link used 224-228
with prepopulated fields, Log A Call button creating from 89-94
Account credit score
rendering graphically, CSS used 62, 63
rendering graphically, Google image chart used 66-68
rendering graphically, JavaScript used 62, 63
rendering graphically, Visualforce used 62-64
Account Detail page 53, 71
Account Google Map
about 244
displaying, within account record 237-241
account name change
by non-system administrators, stopping with validation rule 138-140
Account News custom link 227
Account Revenue field 47
account revenue indicator
building, custom images used 46-53
building, formula field used 46-53
Account Validation Rule Edit page 143, 145
AJAX Toolkit 186
approval assignments
e-mail template, creating for use with 154, 155
Approval History list 170
approval process
about 153
building 154
user setting up, to be associated with 156, 157
Approve button 172
asynchronous call 186

B

BCC field 190
Browse... button 177
Button or Link URL textbox 232

C

Cache Control 177
calendar pop ups
historic year range populating from 1920, for date fields 82-89
year range extending to 2025, for date fields 76-82
Calendar Year Extender checkbox 80
callback function 186
Cascading Style Sheets (CSS)
about 19
used, for rendering Account credit score graphically 62-64

case priority flags
displaying, formula field used 44-46
displaying, salesforce.com images used 44-46
chatter feeds
automatically collapsing, on home page 20-24
on home page, removing 25-29
Close Date field 106, 134
clutter-free Account tab
creating, Visualforce used 54-57
Component Position option 17, 39
confirm() method 186
Contact Filters section 202
Context-Sensitive Help Setting option 179
Create New Approval Process button 159
Credit Score Graphic image 72
CRM Books 234
custom images
used, for building account revenue indicator 46-53
custom link
on sidebar, using to create Personal Setup link 7-9
used for providing Google News search, for account 224-228
using, to open training in new window from sidebar 11-15
custom list button
used, for building mass delete opportunity button 180-187
CustomObjectHelpText 179
custom objects
help pages generating for, PDF document used 177-180
help pages generating for, Visualforce used 177-180

D

Data Settings section 198
date fields
historic year range populating from 1920, on calendar pop ups 82-89
year range extending to 2025, on calendar pop ups 76-82
deals
greater than USD 100,000, opportunity approval process creating for 158-169
Description field 51, 155
Detail Page Button option 91
discover what's new in this release button
on home page, removing 99-104
Display in new window option 11

E

Edit Field Mappings link 200
Email Body field 156
e-mail template
creating, for use with approval assignments 154, 155
Email to Salesforce
enabling, in Salesforce CRM 190-192
Error Location option 139, 144, 146, 150
Error Message section 139
external web application
displayed, by creating custom Web Tab 229-233

F

FamFamFam
URL 47
field update workflow
used, for generating opportunity close date 129-134
used, for setting default opportunity name 122-128
formula
option 50, 175
used, for building account revenue indicator 46-53
used for calculating week number, for opportunity close date 108-111
used, for converting 15 digit Salesforce opportunity ID, to 18 digits 174-176
used for deriving year and month values, from opportunity close date 106-108
used, for displaying case priority flags 44-46

G

geocoding 242
Get Record Count button 203
Google Chart
URL 73
Google image chart
used, for rendering Account credit score graphically 66-68
Google Map
used, for displaying organization location 234-237
Google Map code, options
mapTypeControl 242
mapTypeId 242
zoom 242
Google News search
providing for account, custom link used 224-228
Google-o-meter chart
about 66
URL 66

H

hack 75
Help for this Page link 178, 180
help pages
generating for custom objects, PDF document used 177-180
generating for custom objects, Visualforce used 177-180
Hide Chatter button 21
home page
discover what's new in this release button, removing 99-104
integrated view of cases, displaying 94-99
news-ticker message, creating on 16-19
Opportunity Sales Stages, displaying 34-41
Send An Email button, adding 29-33
Home Page Chatter feeds
automatic collapse, steps for 20-24
removing 25-29
HTML Area option 17, 22

I

integrated view of cases
displaying, in home page 94-99

J

JavaScript
used, for rendering Account credit score graphically 62-64

L

Log A Call button
creating, from accounts with prepopulated fields 89-94

M

mapTypeControl 242
mapTypeId 242
marquee tag
URL 20
mass delete opportunity button
building, custom list button used 180-187
My Unresolved Items page 199

N

New button 16, 30
New Custom Component wizard 98
news-ticker message
creating, on home page 16-19
Next Automated Approver Determined By pick list 161
Next button 16, 84
non-system administrators
stopping, from changing account names with validation rule 138-140

O

opportunity
Reason Lost field creating, validation rule used 148-152
opportunity approval process
building, for deals greater than USD 100,000 158-169

opportunity close date
generating, field update workflow used 129-134
week number calculating, formula used 108-111
year and month values deriving, formula used 106-108
opportunity close date e-mail alert
creating, workflow used 111-122
opportunity name
setting, field update workflow used 122-128
opportunity page layout related
approval history, adding 169
Opportunity Sales Stages
displaying, on home page 34-41
organization location
displaying, Google Map used 234-237
displaying, Visualforce page used 234-237
Google Map code, options 242
Override With option 56

P

PDF document
used, for generating help pages for custom objects 177-180
Personal Setup link
creating, standard Custom Links on sidebar used 7-9
prepopulated fields
Log A Call button, creating from accounts 89-94
productivity tools
for administration 173
for advanced users (superusers) 173
Protected Component option 11

R

Reason Lost field
creating on opportunity, validation rule used 148-152
Record Editability Properties 162
REGEX
UK postcodes accuracy checking, by creating validation rule 145-147
US zip codes accuracy checking, by creating validation rule 143-145

S

Salesforce
historic year range populating from 1920 on calendar pop ups, for data fields 82-89
year range extending to 2025 on calendar pop ups, for data fields 76-82
salesforce.com images
used, for displaying case priority flags 44-46
Salesforce CRM
about 105
data quality, improving 137
Email to Salesforce, enabling 190-192
home tab 5, 6
Salesforce for Outlook configurations, setting up 196-203
Salesforce for Outlook
configurations, setting within Salesforce CRM 196-203
software, configuring on local machine 213-220
software, installing on local machine 204-213
system requisites, obtaining 194, 195
Salesforce for Outlook software. *See* **Salesforce for Outlook**
Salesforce opportunity ID
15 digits converting to 18 digits, formula field used 174-176
Send An Email button
adding, on home page 29-33
Specify Initial Submitters page 164
Static Resources setup screen 177
Subject field 155
Submit for Approval button 170
synchronous call 186

T

tabbed Account Detail page
displaying, Visualforce used 57-61
Template Unique Name field 112
building 155
Training
opening in new window from sidebar, Custom Links used 11-15
Treat blank fields as blanks option 52

two-letter account country code usage
enforcing, validation rule used 140, 142

U

UK postcodes accuracy
checking, by creating validation rule 145-147
Understanding Custom Components Splash screen 22
Unique Name field 133, 160, 165
user
setting up, to be associated with approval process 156, 157
US zip codes accuracy
checking, by creating validation rule 143-145

V

validation rule
creating, to check Uk postcodes accuracy 145-147
creating, to check US zip codes accuracy 143-145
used, for creating Reason Lost field on opportunity 148-152
used for enforcing, two-letter account country code usage 140, 142
used for stopping account name change, by non-system administrators 138-140
Visualforce
page, used for displaying organization location 234-237
page creating, steps for 66
used, for creating clutter-free Account tab 54-57
used, for displaying tabbed Account Detail page 57-61
used, for generating help pages for custom objects 177-180
used, for rendering Account credit score graphically 62-64

W

Web Tab
creating, to display external web application 228-233
week number
calculating for opportunity close date, formula used 108-111
Wide (Right) Column option 17, 22, 101
window.confirm() method 186
workflow
used, for creating opportunity close date e-mail alert 111-122

Y

year and month values
deriving from opportunity close date, formula used 106-108
year range
extending to 2025 on calendar pop ups, for date fields 76-82
populating from 1920 on calendar pop ups, for date fields 82-89

Z

zoom 242

Thank you for buying

Salesforce CRM Admin Cookbook

About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: `www.PacktPub.com`.

About Packt Enterprise

In 2010, Packt launched two new brands, Packt Enterprise and Packt Open Source, in order to continue its focus on specialization. This book is part of the Packt Enterprise brand, home to books published on enterprise software – software created by major vendors, including (but not limited to) IBM, Microsoft and Oracle, often for use in other corporations. Its titles will offer information relevant to a range of users of this software, including administrators, developers, architects, and end users.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to `author@packtpub.com`. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.