Working with OpenERP

Learn to utilize OpenERP to transform and streamline your business

Greg Moss
Working with OpenERP

Learn to utilize OpenERP to transform and streamline your business

Greg Moss
Credits

Author
Greg Moss

Reviewers
Robert Baumgartner
Mantavya Gajjar
Daniel Reis
Andy Skipper

Acquisition Editors
Amarabha Banerjee
Anthony Lowe
Mohammad Rizvi

Lead Technical Editor
Dayan Hyames

Technical Editors
Pooja Arondekar
Amit Ramadas
Amit Shetty

Project Coordinator
Leena Purkait

Proofreaders
Maria Gould
Ameesha Green
Paul Hindle

Indexer
Tejal R. Soni

Production Coordinator
Nitesh Thakur

Cover Work
Nitesh Thakur

www.allitebooks.com
Greg Moss has been a Business and Information Systems Consultant for over 25 years. Starting in 1988, Greg began to work extensively in financial- and accounting-related applications. He wrote his first custom billing system for a rehabilitation facility at the age of 20. He has worked extensively in the health care, point of sale, manufacturing, telecommunications, and service industries. Greg is both a Certified Information Systems Auditor (CISA), a Certified Six Sigma Black Belt, and was the Chief Information Officer at Crownline Boats, Inc.

In addition to OpenERP, he has experience in a variety of ERP systems and was a Sage Pro partner for several years. Greg is the CEO of First Class Ventures, LLC and owner of FirstClassComputerConsulting.com, an OpenERP ready partner.

In his spare time you can find him playing trumpet with his band at a local club, taking cross-country road trips with his African Grey Parrot named Bibi, or sieging a castle with his MMORPG friends.

I would like to thank my wonderful wife Kelly for all her love and support. Also Bibi, for being so sweet and making me laugh every day. Many thanks to the great team at Silkworm for their hard work and dedication.
About the Reviewers

Robert Baumgartner has a degree in Business Informatics from Austria, Europe, where he is living today. He began his career in 2002 as a Business Intelligence Consultant working for different service companies. After this he was working in the paper industry sector as a Consultant and Project Manager for an Enterprise Resource Planning (ERP) system. In 2009, he founded his company datenpol—a service integrator specialist in selected open source software products focusing on business intelligence and ERP. Robert is an open source enthusiast who has given several speeches at open source events. The products he is working on are OpenERP, Talend Data Integration, and JasperReports. He is contributing to the open source community by sharing his knowledge with blog entries on the company blog, http://www.datenpol.at/blog, and he commits software to GitHub such as the OpenERP Talend Connector component which can be found at https://github.com/baumgaro/OpenERP-Talend-Component.

Mantavya Gajjar completed his Bachelor's and Master's of Computing degree in 2003 and 2006, respectively. Since 2006, he has been working as a Director of Tiny ERP Private Limited, a division of OpenERP SA in India. He played a key role in building a strong development team from 1 to more than 100 people at OpenERP's Indian branch.

Tiny ERP Private Limited represents OpenERP SA in India. Tiny ERP (formally known as OpenERP India) was established on 15th March 2007 in India. The aim of Tiny ERP was to set up first level of support center besides the research and development center for OpenERP SA. They have their development center at Infocity, Gandhinagar, India. Tiny ERP is also interested in the growth of business within Asian countries. They have more than 180 employees worldwide, and out of that they have 100 engineers at their Indian office.

Thanks to the author, Greg Moss, for providing this book to OpenERP's community.
Daniel Reis is an IT manager at Securitas, a global security services company, driving the integration of software applications with the company’s operations and processes. This includes initiatives such as OpenERP implementation, enterprise data integration, business intelligence, and SOA/ESB architecture.

He is an Applied Mathematics graduate and MBA, and formerly was Managing Consultant at Capgemini, involved in IT projects for some of the largest Portuguese companies.

He has been an active participant in the OpenERP community for some years, and held several talks on the OpenERP Days official event.

Andy Skipper is a serial startup CTO and Agile Technologist in digital agencies across the United Kingdom. Andy has managed agile teams of developers since 2005 on a variety of technical platforms including Python, PHP, ASP.NET, and J2EE. Most recently, he was the founding CTO at Made.com, where the selection of OpenERP as the organization's back-office system enabled the company to automate key functions and business processes, and to grow explosively to over 100 people and two offices over the space of three years.

Andy lives in Wimbledon, London, with his partner Abby and strapping young son, Toby.
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.


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.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>1</td>
</tr>
<tr>
<td><strong>Chapter 1: Setting Up OpenERP</strong></td>
<td>17</td>
</tr>
<tr>
<td>Getting started with OpenERP online</td>
<td>17</td>
</tr>
<tr>
<td>Taking advantage of OpenERP online</td>
<td>18</td>
</tr>
<tr>
<td>OpenERP browser requirements</td>
<td>18</td>
</tr>
<tr>
<td>Accessing the OpenERP free online trial</td>
<td>18</td>
</tr>
<tr>
<td>Choosing your first application</td>
<td>21</td>
</tr>
<tr>
<td>Configuring accounting data</td>
<td>22</td>
</tr>
<tr>
<td>Setting your accounting options</td>
<td>23</td>
</tr>
<tr>
<td>Signing up for continued service</td>
<td>25</td>
</tr>
<tr>
<td><strong>Using OpenERP without subscription fees</strong></td>
<td>26</td>
</tr>
<tr>
<td>Getting to know the OpenERP architecture</td>
<td>26</td>
</tr>
<tr>
<td>Introducing the PostgreSQL database</td>
<td>26</td>
</tr>
<tr>
<td>Observing database specifications</td>
<td>26</td>
</tr>
<tr>
<td>Running cross platform</td>
<td>27</td>
</tr>
<tr>
<td>Writing code with Python</td>
<td>27</td>
</tr>
<tr>
<td>Following the Model-View-Controller design</td>
<td>27</td>
</tr>
<tr>
<td>Designing models</td>
<td>28</td>
</tr>
<tr>
<td>Rendering views</td>
<td>28</td>
</tr>
<tr>
<td>Authoring controllers</td>
<td>28</td>
</tr>
<tr>
<td><strong>Choosing your installation operating system</strong></td>
<td>28</td>
</tr>
<tr>
<td>Choosing a Microsoft Windows OpenERP installation</td>
<td>29</td>
</tr>
<tr>
<td>Choosing an Ubuntu OpenERP installation</td>
<td>29</td>
</tr>
<tr>
<td>Ubuntu is the primary target platform</td>
<td>29</td>
</tr>
<tr>
<td>Ubuntu is open source</td>
<td>30</td>
</tr>
<tr>
<td>Ubuntu has additional scalability options</td>
<td>30</td>
</tr>
<tr>
<td>Ubuntu has strong community support for OpenERP</td>
<td>30</td>
</tr>
<tr>
<td>Choosing another OS option for OpenERP</td>
<td>31</td>
</tr>
</tbody>
</table>

www.allitebooks.com
<table>
<thead>
<tr>
<th>Understanding OpenERP releases</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrading OpenERP</td>
<td>31</td>
</tr>
<tr>
<td>Installing OpenERP on Windows OS</td>
<td>32</td>
</tr>
<tr>
<td>Performing an all-in-one OpenERP installation in Windows</td>
<td>35</td>
</tr>
<tr>
<td>Testing your Windows installation</td>
<td>39</td>
</tr>
<tr>
<td>Installing OpenERP on Ubuntu</td>
<td>40</td>
</tr>
<tr>
<td>Modifying sources.list</td>
<td>40</td>
</tr>
<tr>
<td>Installing the package</td>
<td>40</td>
</tr>
<tr>
<td>Testing your OpenERP installation</td>
<td>40</td>
</tr>
<tr>
<td>Troubleshooting and OpenERP management tips</td>
<td>40</td>
</tr>
<tr>
<td>Troubleshooting OpenERP installations</td>
<td>41</td>
</tr>
<tr>
<td>Checking your browser destination</td>
<td>41</td>
</tr>
<tr>
<td>Verifying that the OpenERP service is running</td>
<td>41</td>
</tr>
<tr>
<td>Checking for the OpenERP services running in Windows</td>
<td>41</td>
</tr>
<tr>
<td>Checking for OpenERP services running in Ubuntu</td>
<td>42</td>
</tr>
<tr>
<td>Starting and stopping OpenERP services in Ubuntu</td>
<td>42</td>
</tr>
<tr>
<td>Finding the primary OpenERP logfile</td>
<td>42</td>
</tr>
<tr>
<td>Modifying the OpenERP configuration file</td>
<td>43</td>
</tr>
<tr>
<td>Summary</td>
<td>43</td>
</tr>
<tr>
<td>Chapter 2: Starting Your First Company</td>
<td>45</td>
</tr>
<tr>
<td>Gathering requirements</td>
<td>45</td>
</tr>
<tr>
<td>Implementing an ERP system with a modular approach</td>
<td>46</td>
</tr>
<tr>
<td>Introducing Silkworm – our real-world case study</td>
<td>46</td>
</tr>
<tr>
<td>Creating a new database in OpenERP</td>
<td>46</td>
</tr>
<tr>
<td>Setting our master password</td>
<td>47</td>
</tr>
<tr>
<td>Selecting a database name</td>
<td>47</td>
</tr>
<tr>
<td>Loading the demonstration data</td>
<td>48</td>
</tr>
<tr>
<td>Specifying our default language</td>
<td>48</td>
</tr>
<tr>
<td>Choosing a password</td>
<td>48</td>
</tr>
<tr>
<td>Managing databases in OpenERP</td>
<td>49</td>
</tr>
<tr>
<td>Installing the Sales Management module</td>
<td>49</td>
</tr>
<tr>
<td>Configuring accounting data</td>
<td>50</td>
</tr>
<tr>
<td>Knowing the basic OpenERP interface</td>
<td>51</td>
</tr>
<tr>
<td>Using the search box</td>
<td>51</td>
</tr>
<tr>
<td>Picking different views</td>
<td>52</td>
</tr>
<tr>
<td>Creating your first customer</td>
<td>52</td>
</tr>
<tr>
<td>Is a Company?</td>
<td>54</td>
</tr>
<tr>
<td>Entering data into an OpenERP form</td>
<td>54</td>
</tr>
<tr>
<td>Editing customer information – the Sales &amp; Purchases page</td>
<td>55</td>
</tr>
<tr>
<td>Editing customer information – the Accounting page</td>
<td>57</td>
</tr>
<tr>
<td>Entering a product in OpenERP</td>
<td>60</td>
</tr>
<tr>
<td>Product Name</td>
<td>61</td>
</tr>
<tr>
<td>Category</td>
<td>61</td>
</tr>
<tr>
<td>Can be Sold</td>
<td>62</td>
</tr>
<tr>
<td>Entering a product – the Information page</td>
<td>62</td>
</tr>
<tr>
<td>Chapter 3: Exploring Customer Relationship Management in OpenERP</td>
<td>77</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>----</td>
</tr>
<tr>
<td>Using CRM as a business strategy</td>
<td>77</td>
</tr>
<tr>
<td>Managing the customer experience</td>
<td>78</td>
</tr>
<tr>
<td>Treating your customer like a real person</td>
<td>78</td>
</tr>
<tr>
<td>Using your mission statements and company goals to drive the design of your CRM system</td>
<td>79</td>
</tr>
<tr>
<td>Real-world case study – improving customer experience</td>
<td>79</td>
</tr>
<tr>
<td>Assigning the sales representative or account manager</td>
<td>80</td>
</tr>
<tr>
<td>Installing the CRM application</td>
<td>80</td>
</tr>
<tr>
<td>Tracking leads and opportunities</td>
<td>86</td>
</tr>
<tr>
<td>Creating a new opportunity</td>
<td>86</td>
</tr>
<tr>
<td>The Lead tab</td>
<td>88</td>
</tr>
<tr>
<td>Looking at your opportunities in kanban view</td>
<td>90</td>
</tr>
<tr>
<td>Using OpenERP’s OpenChatter feature</td>
<td>91</td>
</tr>
<tr>
<td>Following a business document</td>
<td>92</td>
</tr>
<tr>
<td>Looking at your scheduled calls</td>
<td>94</td>
</tr>
<tr>
<td>Negotiating the sale</td>
<td>94</td>
</tr>
<tr>
<td>Modifying the stages of the sale</td>
<td>95</td>
</tr>
<tr>
<td>Closing the sale</td>
<td>97</td>
</tr>
<tr>
<td>Converting our opportunity into a quotation</td>
<td>98</td>
</tr>
<tr>
<td>Creating leads in OpenERP</td>
<td>98</td>
</tr>
<tr>
<td>Creating custom stages by the sales team</td>
<td>99</td>
</tr>
</tbody>
</table>

**Summary**

75
# Chapter 4: Purchasing with OpenERP

- Understanding the overall purchasing process
  - Setting up a supplier
  - Setting up warehouse locations
  - Generating quotations and purchase orders
  - Receiving the product
  - Settling the invoice
  - Installing the purchasing application
  - Setting up your first supplier
  - Designating supplier companies versus individuals
  - Configuring your product for procurement
    - Defining the procurement method
    - Knowing your supply method
  - Setting the cost price of the product
    - Estimating manufacturing lead time
    - Setting records to active
  - Assigning suppliers to the product
    - Establishing the supplier
    - Designating supplier product name and product code
    - Setting minimal quantity
    - Calculating the delivery time
  - Creating your first purchase quotation
    - Adding products to your quotation
    - Printing a quotation and updated status
    - Promoting status to RFQ (request for quotation) sent
    - Confirming a purchase order
  - Receiving products
    - Getting ready to receive
    - Receiving our goods
  - Paying supplier invoices

# Summary

# Chapter 5: Making Goods with Manufacturing Resource Planning

- Creating manufacturing orders
  - Producing the product
  - Delivering the order
  - Defining the workflow for your business

A real-world example – producing a custom-printed t-shirt

- Installing manufacturing resource planning (MRP)
- Creating your first manufacturing order
- Building your bill of materials
  - Confirming production
- Producing the product
  - Analyzing the inventory
  - Managing routings and work orders
<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating a work center 129</td>
</tr>
<tr>
<td>Defining the resource type 130</td>
</tr>
<tr>
<td>Creating routing orders 132</td>
</tr>
<tr>
<td>Creating a manufacturing order with routing and work center 133</td>
</tr>
<tr>
<td>Producing the manufacturing order 134</td>
</tr>
<tr>
<td><strong>Summary</strong> 135</td>
</tr>
<tr>
<td><strong>Chapter 6: Configuring Accounting and Finance</strong> 137</td>
</tr>
<tr>
<td>Defining the chart of accounts for your business 137</td>
</tr>
<tr>
<td>Installing the Accounting and Finance application 138</td>
</tr>
<tr>
<td>Viewing the current chart of accounts 139</td>
</tr>
<tr>
<td>How were transactions created in Account Payable? 140</td>
</tr>
<tr>
<td>Viewing the other journal items 141</td>
</tr>
<tr>
<td>Tracking accounting transactions through the sales and accounts receivable process 141</td>
</tr>
<tr>
<td>Viewing the transactions created by validating the invoice 144</td>
</tr>
<tr>
<td>Practice posting transactions and tracking the results 147</td>
</tr>
<tr>
<td>Where are my taxes? 147</td>
</tr>
<tr>
<td>Specifying the account for your sales tax 148</td>
</tr>
<tr>
<td><strong>Setting up your own accounts</strong> 149</td>
</tr>
<tr>
<td>Specifying a new account for your product category 150</td>
</tr>
<tr>
<td>Configuring your fiscal year and periods 152</td>
</tr>
<tr>
<td>Closing a period 153</td>
</tr>
<tr>
<td>Creating journal entries 153</td>
</tr>
<tr>
<td><strong>Summary</strong> 155</td>
</tr>
<tr>
<td><strong>Chapter 7: Implementing the Human Resources Application</strong> 157</td>
</tr>
<tr>
<td>Taking a modular approach to human resources 157</td>
</tr>
<tr>
<td>Installing the Employee Directory application 158</td>
</tr>
<tr>
<td>Creating a new employee 158</td>
</tr>
<tr>
<td>Employees’ public information 159</td>
</tr>
<tr>
<td>Related User 160</td>
</tr>
<tr>
<td>Visibility 160</td>
</tr>
<tr>
<td>Department 160</td>
</tr>
<tr>
<td>Job 161</td>
</tr>
<tr>
<td>Manager and Coach 161</td>
</tr>
<tr>
<td><strong>Managing timesheets</strong> 163</td>
</tr>
<tr>
<td>Looking at your current timesheet 163</td>
</tr>
<tr>
<td>Setting up an account for tracking time 164</td>
</tr>
<tr>
<td>Configuring the employee for entering timesheets 165</td>
</tr>
<tr>
<td>Tracking employee hours with the timesheet 166</td>
</tr>
<tr>
<td>Validating timesheets 167</td>
</tr>
<tr>
<td><strong>Leave management</strong> 168</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Creating a leave request</td>
</tr>
<tr>
<td>Leave Type</td>
</tr>
<tr>
<td>Duration</td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>Employee</td>
</tr>
<tr>
<td>Department</td>
</tr>
<tr>
<td>Submitting for approval</td>
</tr>
<tr>
<td>Approving leave requests</td>
</tr>
<tr>
<td>The Leaves Summary page</td>
</tr>
<tr>
<td><strong>Managing the recruitment process</strong></td>
</tr>
<tr>
<td>Defining recruitment stages</td>
</tr>
<tr>
<td>Recruiting for a new job</td>
</tr>
<tr>
<td>Understanding the Interview button</td>
</tr>
<tr>
<td>Creating an employment application</td>
</tr>
<tr>
<td>Hiring employees</td>
</tr>
<tr>
<td><strong>Working with user access rights and groups</strong></td>
</tr>
<tr>
<td>Knowing built-in access rights</td>
</tr>
<tr>
<td>Understanding inherited groups</td>
</tr>
<tr>
<td>Setting access to menus and views</td>
</tr>
<tr>
<td>Determining access rights</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Chapter 8: Understanding Project Management</strong></td>
</tr>
<tr>
<td>The basics of Project Management</td>
</tr>
<tr>
<td>Installing the Project Management application</td>
</tr>
<tr>
<td>A real-world project example</td>
</tr>
<tr>
<td>Creating our first project</td>
</tr>
<tr>
<td>Assigning project teams</td>
</tr>
<tr>
<td>Setting other project info</td>
</tr>
<tr>
<td>Understanding project stages</td>
</tr>
<tr>
<td>Defining project stages for a specific project</td>
</tr>
<tr>
<td><strong>Defining project tasks</strong></td>
</tr>
<tr>
<td>Defining additional task information</td>
</tr>
<tr>
<td>Creating additional tasks</td>
</tr>
<tr>
<td>Completing project tasks</td>
</tr>
<tr>
<td><strong>Estimating project costs and time</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Chapter 9: Creating Advanced Searches and Dashboards</strong></td>
</tr>
<tr>
<td>Determining the search requirements for your business</td>
</tr>
<tr>
<td>Searching in OpenERP</td>
</tr>
<tr>
<td>Using filters in list views</td>
</tr>
<tr>
<td>Table of Contents</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>Grouping information in lists 205</td>
</tr>
<tr>
<td>Performing an advanced search 207</td>
</tr>
<tr>
<td>Specifying multiple advanced searches 208</td>
</tr>
<tr>
<td>Saving your advanced searches 209</td>
</tr>
<tr>
<td>Adding information to your custom dashboard 210</td>
</tr>
<tr>
<td><strong>Summary</strong> 214</td>
</tr>
<tr>
<td><strong>Chapter 10: Customizing OpenERP for Your Business</strong> 215</td>
</tr>
<tr>
<td>Activating the developer mode 215</td>
</tr>
<tr>
<td>Getting out of the developer mode 217</td>
</tr>
<tr>
<td><strong>Backing up your database</strong> 218</td>
</tr>
<tr>
<td>Restoring a database in OpenERP 219</td>
</tr>
<tr>
<td><strong>Adding a custom field to OpenERP</strong> 220</td>
</tr>
<tr>
<td>Adding another field to the sales order 226</td>
</tr>
<tr>
<td><strong>Adding a field to an existing view</strong> 228</td>
</tr>
<tr>
<td><strong>Summary</strong> 229</td>
</tr>
<tr>
<td><strong>Chapter 11: Understanding Workflows</strong> 231</td>
</tr>
<tr>
<td>Example of OpenERP workflow 232</td>
</tr>
<tr>
<td>Activating the developer mode 232</td>
</tr>
<tr>
<td>Viewing the actual step in the workflow 234</td>
</tr>
<tr>
<td>Getting help from OpenERP 234</td>
</tr>
<tr>
<td><strong>Planning the changes to our workflow</strong> 234</td>
</tr>
<tr>
<td>Looking at an OpenERP workflow in detail 234</td>
</tr>
<tr>
<td>Looking at the Sales Orders workflow inside the workflow designer 235</td>
</tr>
<tr>
<td>Workflow nodes 237</td>
</tr>
<tr>
<td>Looking closer at the sales order workflow 237</td>
</tr>
<tr>
<td>Examining a workflow node in detail 239</td>
</tr>
<tr>
<td>Transitions in workflows 241</td>
</tr>
<tr>
<td><strong>Ending the workflow</strong> 242</td>
</tr>
<tr>
<td><strong>Modifying workflows</strong> 243</td>
</tr>
<tr>
<td>Adding a step into the workflow for art approval 243</td>
</tr>
<tr>
<td>Editing the sales order workflow 243</td>
</tr>
<tr>
<td>Summary of the steps for changing the workflow 244</td>
</tr>
<tr>
<td>Adding a new activity node to a workflow 245</td>
</tr>
<tr>
<td>Modifying the transitions to include art approval 246</td>
</tr>
<tr>
<td>Restricting the art approval to the sales manager 248</td>
</tr>
<tr>
<td><strong>Summary</strong> 253</td>
</tr>
<tr>
<td><strong>Chapter 12: Modifying Documents and Reports</strong> 255</td>
</tr>
<tr>
<td>Company headers and footers 256</td>
</tr>
<tr>
<td>RML – Report Markup Language 257</td>
</tr>
<tr>
<td>Examining the company header and footer 257</td>
</tr>
<tr>
<td>RML for the company header 258</td>
</tr>
</tbody>
</table>
Table of Contents

Making our first simple change 258
RML for the company footer 259
Understanding dynamic fields in your reports 260
Creating a custom footer 260
Modifying OpenERP documents using OpenOffice 262
Installing OpenOffice 262
Installing the OpenERP OpenOffice plugin 262
Troubleshooting the OpenOffice Report Designer installation 264
Modifying the sales order 264
Using OpenOffice Writer to modify the report 266
Making a small change first 266
Saving our changes back to the server 267
Working with dynamic fields in reports 269
Alternative reporting systems 270
OpenERP Webkit Report 270
Jaspersoft reports 271
Translating and reporting data with Pentaho or Kettle 272
Summary 272
Chapter 13: Discovering Custom OpenERP Modules 273
Exploring the OpenERP application and module directory 274
Building our first OpenERP module 274
Specifying a custom directory to hold our OpenERP modules 275
Contents of our module directory 275
Creating and editing the files 276
Extending an OpenERP model in silkworm.py 278
Extending our module to customize the sales order workflow 286
Adding a button to the sales order 286
Modifying the available states of the sales order model 288
Finding your way around the base OpenERP modules 288
Updating the other buttons for the workflow 292
Seeing it all come together 293
Using a module to add a filter to a search view 294
Adding the code to create the rush order filter 295
Creating the filter 296
Summary 296
Appendix: Locating Additional OpenERP Resources 297
Locating the essential OpenERP documentation 297
Visiting the official OpenERP help site 297
Finding OpenERP applications and modules 298
Getting the latest OpenERP 7 release notes 298
Downloading OpenERP from Launchpad 298
Locating resources on Ubuntu 298
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting access to the additional developer documentation</td>
<td>299</td>
</tr>
<tr>
<td>Learning from the latest technical memento</td>
<td>299</td>
</tr>
<tr>
<td>Accessing the OpenERP technical documentation</td>
<td>299</td>
</tr>
<tr>
<td>Getting quick access to OpenERP installations using OpenERP Runbot</td>
<td>299</td>
</tr>
<tr>
<td>Finding Postgres resources</td>
<td>300</td>
</tr>
<tr>
<td>Locating Python resources</td>
<td>300</td>
</tr>
<tr>
<td>Finding XML resources</td>
<td>300</td>
</tr>
<tr>
<td>Locating RML resources</td>
<td>300</td>
</tr>
<tr>
<td><strong>Alternative reporting solutions</strong></td>
<td>301</td>
</tr>
<tr>
<td>Jaspersoft reports</td>
<td>301</td>
</tr>
<tr>
<td>Community edition of JasperReports Server</td>
<td>301</td>
</tr>
<tr>
<td>Community edition of iReport Designer</td>
<td>301</td>
</tr>
<tr>
<td>Pentaho/Kettle</td>
<td>301</td>
</tr>
<tr>
<td>Aeroo Reports</td>
<td>301</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>303</td>
</tr>
</tbody>
</table>
Preface

*Working with OpenERP* provides a comprehensive walkthrough for installing, configuring, and implementing OpenERP in real-world business environments. This book will assist you in understanding the value of *Enterprise Resource Planning* (ERP) systems and best practice approaches for getting a system up and running in your organization. For those that are new to ERP systems this book will serve as a solid introduction. In later chapters the book covers advanced configurations and will show you how to customize OpenERP to fit the needs of practically any business.

**What this book covers**

The book is divided into three sections:

- Installation of OpenERP and the basics for implementing OpenERP in your business (*Chapter 1, Setting Up OpenERP* through *Chapter 4, Purchasing with OpenERP*)

- Introduction to accounting and finance setup and modules to help your business run more efficiently (*Chapter 5, Making Goods with Manufacturing Resource Planning* through *Chapter 8, Understanding Project Management*)

- Advanced configurations and customization of OpenERP (*Chapter 9, Creating Advanced Searches and Dashboards* through *Chapter 12, Modifying Documents and Reports*)

Now let's discuss in some detail what each chapter will cover:

*Chapter 1, Setting up OpenERP*, covers the different installation types and prerequisites for both Windows and Ubuntu. Instruction is provided for finding the right download package and setting up OpenERP. The chapter then goes into the basics for configuring OpenERP. We will also discuss the advantages of running OpenERP in the cloud and the basic sign-up process for accessing a demonstration installation to get an overview of the software.
Chapter 2, Starting Your First Company, begins by introducing you to the real-world case study that will be used as an example throughout the book. We continue by learning how to create the company database and configure the basic company settings required to quickly get your first OpenERP system up and running. The first module, Sales Management, will be installed, and we will walk through the steps of entering a customer and a product. The chapter concludes by entering a sales order and completing the sale.

Chapter 3, Exploring Customer Relationship Management in OpenERP, starts with a basic overview of CRM systems and their importance in today’s modern business environment. After we cover the installation of the CRM module, a lead is entered for our sample company. We will demonstrate CRM workflow by turning the lead into a customer. Next, a quote is generated for our newly acquired customer, and a call is scheduled for follow-up. Finally we give an introduction to social features that can be used to communicate with employees and customers throughout the sales process.

Chapter 4, Purchasing with OpenERP, shows us how to install the purchasing module, set up suppliers and begin purchasing and receiving products in OpenERP. Later in the chapter we learn how to tie purchasing into sales orders to automatically generate draft purchase orders based on your business requirements.

Chapter 5, Making Goods with Manufacturing Resource Planning, begins to explore some of the primary functionality of ERP systems for manufacturing operations. You will learn how to set up your manufacturing orders and define a bill of materials to specify the raw materials that will go into your final products. Manufacturing operations can then be extended with routing and work centers to provide you with more control in tracking time and resources.

Chapter 6, Configuring Accounting and Finance, discusses the Accounts Receivable and Accounts Payable basic functions. Next, we will introduce the chart of accounts and discover how to set up fiscal periods. This chapter will also include the basic accounting reports and how to close a period.

Chapter 7, Implementing the Human Resources Application, begins by installing the basic HR modules and going over the employee directory. Other topics in the chapter will include timesheets, recruitment process, and leave management. The chapter concludes with installing the Document Manager and using it to provide documents to employees.

Chapter 8, Understanding Project Management, covers the features of the Project Management module in OpenERP. We will create a project, see how to enter tasks, and tie a project to a specific customer. Next, team members are assigned to the project, and we configure task stages. We will then go over real-world examples of using the Project Management module to more easily manage complex orders and customer needs with greater ease.
Chapter 9, Creating Advanced Searches and Dashboards, demonstrates how to utilize the advanced search features and configure custom dashboards in OpenERP. By the end of the chapter, the reader will be able to create and save custom searches to reuse later as well as add search results to dashboards.

Chapter 10, Customizing OpenERP for Your Business, explains how to enter developer mode for making a variety of custom changes to OpenERP. We will walk through the steps to add fields to the sales order form and then include the fields in tree views for sorting and reporting. From here, we will get into advanced configuration topics to better customize OpenERP for your specific business requirements.

Chapter 11, Understanding Workflows, introduces the workflow editor and analyzes the basic sales order workflow. Using our case study example, the workflow is modified to improve the flow of information through the business. By the end of the chapter you should have a basic understanding of modifying workflows to better handle unique business processes.

Chapter 12, Modifying Documents and Reports, goes over the basic reporting mechanisms available in OpenERP and weighs the advantages and disadvantages of the various options. We then demonstrate how to install the OpenOffice reporting module and make changes to a report.

Chapter 13, Discovering Custom OpenERP Modules, introduces the process of developing custom solutions in OpenERP. We build on what we have learned in customizing OpenERP and create a module that will persist our custom field and views within our module. Next we build on the workflow modifications we made in Chapter 11, Understanding Workflows, and upgrade our module to approve art designs for our real-world example.

Chapter 14, Integrating Warehouse and Inventory Management Practices, helps us configure our installation when inventory management becomes more sophisticated and involves multiple warehouses and multiple locations. You can download this from https://www.packtpub.com/sites/default/files/downloads/3800OS_Chapter14_Integrating_Warehouse_and_Inventory_Management_Practices.pdf

Appendix, Locating Additional OpenERP Resources, provides a useful collection of links that will help you in deploying an OpenERP installation.
What you need to know to use this book

To get the most out of this book you should have an understanding of basic business operations. For example, you should know the purpose of a sales order and a purchase order. The reader should also have basic computer skills for understanding file systems and installing software. For more advanced customization topics in the book, the reader should have a basic knowledge of databases and programming concepts.

Who this book is for

This book is for anyone that is interested in implementing an ERP system in a business organization. If you are an IT professional looking to get a functional understanding of OpenERP, then this book is for you. This book is also appropriate business and operations managers who wish to get a comprehensive understanding of OpenERP and how it can be used to improve business processes.

What is an ERP system?

An Enterprise Resource Planning (ERP) system is essentially a suite of business applications that are integrated together to assist a company in collecting, managing, and reporting information throughout core business processes. These business applications, typically called modules, can often be independently installed and configured based on the specific needs of the business. As the needs of the business change and grow, additional modules can be incorporated into an existing ERP system to better handle the new business requirements. This modular design of most ERP systems gives companies great flexibility in how they implement the system.

In the past, ERP systems were primarily utilized in manufacturing operations. Over the years, the scope of ERP systems has grown to encompass a wide range of business-related functions. Recently ERP systems have started to include more sophisticated communication and social networking features.

Common ERP modules

The core applications of an ERP system typically include:

- Sales Orders
- Purchase orders
- Accounting and finance
Manufacturing Resource Planning (MRP)
Customer Relationship Management (CRM)
Human Resources (HR)

Let us take a brief look at each of these modules and how they address specific business needs.

Selling products to your customer
Sales Orders (SO), are documents that a business generates when they sell products and services to a customer. In an ERP system, the Sales Order module usually will allow management of customers and products to optimize efficiency for data entry of the sales order. Many sales orders begin as customer quotes. Quotes allow a salesperson to collect order information that may change as the customer makes decisions on what they want in their final order.

Once a customer has decided exactly what they wish to purchase, the quote is turned into a sales order and is confirmed for processing. Depending on the requirements of the business, there are a variety of methods to determine when a customer is invoiced or billed for the order.

The preceding screenshot shows a sample sales order in OpenERP.
Purchasing products from suppliers

Purchase Orders, often known as PO, are documents that a business generates when they purchase products from a vendor. The Purchase Order module in an ERP system will typically include management of vendors (also called suppliers) as well as management of the products that vendor carries. Much like sales order quotes, a purchase order system will allow a purchasing department to create draft purchase orders before they are finalized into a specific purchasing request.

Often a business will configure the Sales Order and Purchase Order modules to work together to streamline business operations. When a valid sales order is entered, most ERP systems will allow you to configure the system so that a purchase order can be automatically generated if the required products are not in stock to fulfill the sales order. ERP systems will allow you to set minimum quantities on-hand or order limits that will automatically generate purchase orders when inventory falls below a predetermined level. When properly configured, a purchase order system can save a significant amount of time in purchasing operations and assist in preventing supply shortages.

The preceding screenshot shows a sample Purchase Order in OpenERP.
Accounting and finance

Accounting and finance modules integrate with an ERP system to organize and report business transactions. In many ERP systems, the accounting and finance module is known as GL for General Ledger. All accounting and finance modules are built around a structure known as the chart of accounts. The chart of accounts organizes groups of transactions into categories such as assets, liabilities, income, and expenses. ERP systems provide a lot of flexibility in defining the structure of your chart of accounts to meet the specific requirements for your business.

Accounting transactions are grouped by date into periods (typically by month) for reporting purposes. These reports are most often known as financial statements. Common financial statements include balance sheets, income statements, cash flow statements, and statements of owner's equity.

Managing your accounts and financing in OpenERP

Accounting and finance modules integrate with an ERP system to organize and report business transactions. In many ERP systems, the accounting and finance module is known as GL for General Ledger. All accounting and finance modules are built around a structure known as the chart of accounts. The chart of accounts organizes groups of transactions into categories such as assets, liabilities, income, and expenses. ERP systems provide a lot of flexibility in defining the structure of your chart of accounts to meet the specific requirements for your business.

Accounting transactions are grouped by date into periods (typically by month) for reporting purposes. These reports are most often known as financial statements. Common financial statements include balance sheets, income statements, cash flow statements, and statements of owner's equity.

Handling your manufacturing operations

The Manufacturing Resource Planning (MRP) module manages all the various business operations that go into the manufacturing of products. The fundamental transaction of an MRP module is a manufacturing order, which is also known as a production order in some ERP systems. A manufacturing order describes the raw products or subcomponents, steps, and routings required to produce a finished product. The raw products or subcomponents required to produce the finished product are typically broken down into a detailed list called a bill of materials or BOM. A BOM describes the exact quantities required of each component and are often used to define the raw material costs that go into manufacturing the final products for a company.
Often an MRP module will incorporate several submodules that are necessary to define all required operations. Warehouse management is used to define locations and sublocations to store materials and products as they move through the various manufacturing operations. For example, you may receive raw materials in one warehouse location, assemble those raw materials into subcomponents and store them in another location, then ultimately manufacture the end products, and store them in a final location before delivery to the customer.

**Managing customer relations in OpenERP**

In today's business environment, quality customer service is essential to being competitive in most markets. A Customer Relationship Management (CRM) module assists a business in better handling the interactions they may have with each customer. Most CRM systems also incorporate a presales component that will manage opportunities, leads, and various marketing campaigns.

Typically, a CRM system is utilized the most by the sales and marketing departments within a company. For this reason, CRM systems are often considered to be sales force automation tools or SFA tools. Sales personnel can set up appointments, schedule call backs, and employ tools to manage their communications. More modern CRM systems have started to incorporate social networking features to assist sales personnel in utilizing these newly emerging technologies.

**Configuring human resource applications in OpenERP**

Human Resource modules, commonly known as HR, manage the workforce or employee-related information in a business. Some of the processes ordinarily covered by HR systems are payroll, time and attendance, benefits administration, recruitment, and knowledge management.
Increased regulations and complexities in payroll and benefits have led to HR modules becoming a major component of most ERP systems. Modern HR modules typically include employee "kiosk" functions to allow employees to self-administer many tasks such as putting in a leave request or checking on their available vacation time.

**Finding additional modules for your business requirements**

In addition to core ERP modules, OpenERP has many more official and community-developed modules available. At the time of this book's publication, the OpenERP application repository had 1,348 modules listed for Version 7! Many of these modules provide small enhancements to improve usability, like adding payment type to a sales order. Other modules offer e-commerce integration or complete application solutions, such as managing a school or hospital.

Here is a short list of the more common modules you may wish to include in an OpenERP installation:

- Point of Sale
- Project Management
- Analytic Accounting
- Document Management System
- Outlook Plugin
- Country-Specific Accounting Templates
- OpenOffice Report Designer
Throughout the book you will be introduced to various OpenERP modules that extend the functionality of the base OpenERP system. You can find a complete list of OpenERP modules at: http://v6apps.openerp.com/.

The preceding screenshot shows the module selection page in OpenERP.

**Getting into OpenERP**

Do you want to jump in right now and get a look at OpenERP 7 without any complex installations? Well, you are in luck! You can access an online installation of OpenERP where you can get a peek at many of the core modules right from your web browser. The installation is shared publicly, so you will not want to use this for any sensitive information. It is ideal, however, for getting a quick overview of the software and for getting an idea for how the interface functions.

You can access the demonstration version of OpenERP at: https://demo2.openerp.com.
OpenERP – an open source ERP solution

OpenERP is a collection of business applications that are available under an open source license. For this reason, OpenERP can be used without paying license fees and can be customized to suit the specific needs of a business. There are many advantages to open source software solutions. Some of these advantages are covered in the following sections.

Free your company from expensive software license fees

One of the primary downsides of most ERP systems is they often involve expensive license fees. Increasingly, companies must pay these license fees on an annual basis just to receive bug fixes and product updates. Because ERP systems can require companies to devote great amounts of time and money for setup, data conversion, integration, and training, it can be very expensive, often prohibitively so, to change ERP systems. For this reason many companies feel trapped as their current ERP vendors increase license fees.

Choosing open source software solutions frees a company from the real possibility that a vendor will increase license fees in the years ahead.

Modify the software to meet your business needs

With proprietary ERP solutions, you are often forced to accept the software solution the vendor provides chiefly "as is". While you may have customization options and can sometimes pay the company to make specific changes, you rarely have the freedom to make changes directly to the source code yourself. The advantages to having the source code available to enterprise companies can be very significant. In a highly competitive market being able to develop solutions that improve business processes and give your company the flexibility to meet future demands can make all the difference.
Collaborative development

Open source software does not rely on a group of developers who work secretly to write proprietary code. Instead, developers from all around the world work together transparently to develop modules, prepare bug fixes, and increase software usability. In the case of OpenERP, all of the source code is available at https://launchpad.net/. Here developers submit their code changes through a structure called branches. Changes can be peer reviewed, and once the changes are approved, they are incorporated into the final source code product.

OpenERP – AGPL Open Source License

The term "open source" covers a wide range of open source licenses that have their own specific rights and limitations. OpenERP and all of its modules are released under the Affero General Public License (AGPL) version 3. One key feature of this license is that any custom developed module running under OpenERP must be released with source code. This stipulation protects the OpenERP community as a whole from developers who may have a desire to hide their code from everyone else.

You can find the full AGPL license at: http://www.gnu.org/licenses/agpl-3.0.html.

A real-world case study using OpenERP

The goal of this book is to do more than just walk through the various screens and reports of OpenERP. Instead we want to give you a solid understanding of how you would implement OpenERP to solve real-world business problems. For this reason, this book will present a real-life case study in which OpenERP was actually utilized to improve specific business processes.

Silkworm, Inc. – a mid-sized screen printing company

Silkworm, Inc. is a mid-sized silkscreen printer in the Midwest that manufactures and sells a variety of custom apparel products. Using OpenERP, we will set up the company records (or system) from scratch and begin by walking through their most basic sales order process: selling t-shirts. From there, we will move on to manufacturing operations where custom art designs are developed and then screen printed onto raw materials for shipment to customers.
The marketing and sales departments at Silkworm rely heavily on the CRM functions in OpenERP. We will go over how they manage opportunities and leads and take advantage of the many communication features of OpenERP. The project manager module plays a central role in organizing complex projects that can involve multiple sales orders and detailed product requirements.

In later chapters, we will get into some of the OpenERP customization that was required to meet the specific needs of our case study's business. New fields to collect data and changes in workflow are examples of the customization we will be covering in this real-life example. Finally, we will introduce more complex custom development concepts that call for (or necessitate) modifying the actual source code of OpenERP to streamline business processes and improve efficiency for the business.

**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, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "We could create our directory in the addons folder."

A block of code is set as follows:

```xml
<field name="x_daterquired"/>
<field name="x_rush"/>
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```python
'th_weight': fields.float('Weight', readonly=True,
states={('draft': [{('readonly', False)}]},

'state': fields.selection([('cancel', 'Cancelled'), ('draft',
'Draft'), ('confirmed', 'Confirmed'), ('exception',
'Exception'), ('done', 'Done')], 'Status', required=True,
readonly=True,)
```

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: "You can click on the **Skip this step** link to go ahead and begin using OpenERP immediately."
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/submit-errata, 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.
Setting Up OpenERP

OpenERP is a powerful set of open source business applications built on the OpenObject framework. When you first install OpenERP the only functionality you will have is limited messaging options between users. From there, OpenERP allows you to install exactly the modules you need, as you need them. This flexibility makes OpenERP much more accessible than many ERP solutions.

In this chapter we will start working with OpenERP by covering the installation and basics of setting up an OpenERP database.

The topics we cover will include:

- How to get started with running OpenERP online
- Setting up a trial company
- Installing OpenERP in Windows and Ubuntu
- Troubleshooting and configuring your installation

Getting started with OpenERP online

It was not long ago that nearly all companies kept their primary information systems in-house. This approach requires not just a lot of capital expense in purchasing servers and software licenses, it also creates a lot of responsibility and risk in backing up data and ensuring business continuity. Today, more and more companies are choosing to host their business applications on online networks commonly known as the cloud. OpenERP allows you the flexibility of both options; either hosting on your own hardware or utilizing OpenERP’s online software services.
Taking advantage of OpenERP online

The best thing about accessing OpenERP online is that you can jump in and start using the software right away. You don't have to decide what operating system to use. You don't have to install any software at all. Just enter the address into your web browser and go.

Another added benefit of taking this approach is you will verify that your web browser is up-to-date and compatible with Version 7 of OpenERP. So even if you intend to install OpenERP on your own hardware, it is still worth taking a minute to test out the online OpenERP trial a bit to make sure you do not have any browser compatibility issues.

OpenERP browser requirements

OpenERP is designed to run on a variety of modern web browsers. Supported browsers include:

- Google Chrome (recommended)
- Internet Explorer
- Firefox
- Safari

Macintosh users will need to make sure they are running Mac OS X or above. Users running older Macintosh systems are currently having difficulties running OpenERP Version 7. Also, in most cases Google Chrome tends to offer the best experience when working with OpenERP.

Accessing the OpenERP free online trial

Accessing the OpenERP free online trial could not be simpler. Just open up your browser and navigate to:

https://www.openerp.com/start
The OpenERP trial site prompts you to supply a domain name that you will use to access your trial version, as well as prompting you to select the language. Choose a name to fill in the **Your Domain** field with a description and make a note of it so you can easily find your trial instance later.

This form will let you choose a name for your trial company. Click on the **Create My Company** button to continue to the next step in setting up your free online trial.
In the next step you are prompted to create an optional account so you can easily extend your trial.

At this point you can go ahead and sign up to create an account for your free trial, or you can click on the **Skip this step** link to go ahead and begin using OpenERP immediately. So to start using OpenERP right away and avoid filling out more forms, we will click on the **skip this step** link to get right into OpenERP.

Once you click on **Start Now**, you are taken to a screen to select the first application you wish to install.
Choosing your first application

OpenERP can be considered as a collection of business applications. Unlike many ERP systems; with OpenERP, you can choose to install exactly the modules that fit your business needs. Perhaps you are interested in starting with sales orders and then later integrating Customer Relationship Management (CRM) for your company. With OpenERP you can begin by just installing Sales Management, adding additional applications as needed.
Setting Up OpenERP

This page will let you choose your first application to install. In the remainder of the book we will go through many of these applications in detail. For this example, we are going to install the Sales Management application.

Click on the Install button under Sales Management to install your first OpenERP module.

Configuring accounting data

When you initially install any module in OpenERP that deals with transactions or currency, you will be prompted to configure accounting data. The first step in this process is to select Accounting Package for your company.

Technically, you are not really installing an accounting package but are choosing the localized chart of accounts you wish to use for the application. OpenERP offers localization for more than 25 countries.

For most installations, this will coincide with the country in which your company primarily does business.

We leave Company selected as Your Company and select Continue to go on to the next form in the process.
Setting your accounting options

After selecting your accounting package, you will be prompted to set up various accounting options.

This form lets you pick a template for your chart of accounts. There are various options available depending on your business type.

For now you can just leave this as **Configurable Account Chart Template**. Next you will select your currency and set your sales and purchase tax rates. Remember, this is just a demonstration to get you started with OpenERP. Later we will go into detail on the various options available.
Setting Up OpenERP

After clicking on **Apply**, your trial version will become active and you can begin working with your new OpenERP installation. Since we have installed the sales management application we are taken to the customer screen where we can begin creating new customers for our demonstration company.

Take a few minutes to look around in OpenERP to get familiar with the interface. You don't have to worry about breaking anything or doing anything wrong. If you run into problems or get confused, just close your web browser and try again.

This is a demonstration and will only last for one 4-hour session. If you close your browser, you will lose your setup and will have to start over again.
Sign up for continued service

As you should see, it is very easy to get started with running OpenERP online. Keep in mind though, that everything you are doing in OpenERP up to this point will not be saved after your current session. However, it is easy to sign up to continue using OpenERP online.

Running OpenERP online is not free. Currently, it will cost you $39.00 a month per user to subscribe. For that $39.00 monthly fee, according to the OpenERP subscription page, you are entitled to:

- Installation of any modules in one click
- Automated updates for bug fixes
- Upgrade service to benefit from new features
- Unlimited bug-fix guarantee
- Hosting (optional)
- Support (2 hours)

Depending on your requirements, an OpenERP subscription may be a good decision. Installing and maintaining an OpenERP installation takes a degree of expertise and has risks for production systems. You must maintain adequate disaster recovery procedures in case of server crashes or hard drive failures. There are also complexities in applying bug fixes and migrating to newer versions of OpenERP. This book will help you with many of these tasks. Still, it can be quite convenient to have an OpenERP subscription so you can focus on the functional rather than the technical aspects of working with OpenERP.

If you decide you wish to pay for the convenience of running OpenERP online, click on **Register now to add 15 days for free!** at the top of the page.

You will then be taken to the same sign-up sheet you were presented with after you provided the name of your company.
Using OpenERP without subscription fees
If you choose not to pay the subscription fee, do not fear! The remainder of this chapter will assist you with installing OpenERP on your own hardware.

Getting to know the OpenERP architecture
Setting up and managing an OpenERP installation will require a basic understanding of the components that make up OpenERP. Every ERP system has a set of technologies and underlying software platforms that are required for the system to function. Fortunately, unless you plan to do OpenERP customization, you only need to understand the very basics of the OpenERP architecture to manage the installation.

In this book we provide a basic overview of the OpenERP architecture. If you wish to get more detailed documentation on the OpenERP architecture visit:
https://doc.openerp.com/trunk/server/02_architecture/

Introducing the PostgreSQL database
Like most ERP systems, OpenERP has specific database requirements. In this case, that is PostgreSQL. PostgreSQL is an open source, cross-platform object relational database management system (ORDMS). While not popular on the scale of Microsoft SQL Server or MySQL, PostgreSQL is an enterprise-class database server with many advanced features. In fact, PostgreSQL stacks up very well against far more expensive databases such as Microsoft SQL Server and Oracle Database.

Observing database specifications
From a specification standpoint, PostgreSQL is perfectly capable of handling nearly any OpenERP installation. In most instances you will hit limits of OpenERP scalability and performance long before you will go past the hard limits of PostgreSQL itself. Still, when installing any ERP system, it is important that you clearly understand the most critical database specifications. Here are the specs for PostgreSQL:
### Limit Value

<table>
<thead>
<tr>
<th>Limit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum database size</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Maximum table size</td>
<td>32 TB</td>
</tr>
<tr>
<td>Maximum row size</td>
<td>1.6 TB</td>
</tr>
<tr>
<td>Maximum field size</td>
<td>1 GB</td>
</tr>
<tr>
<td>Maximum rows per table</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Maximum columns per table</td>
<td>250 - 1600 depending on column types</td>
</tr>
<tr>
<td>Maximum indexes per table</td>
<td>Unlimited</td>
</tr>
</tbody>
</table>

### Running cross platform

PostgreSQL runs on every major operating system. For most OpenERP installations, Ubuntu is the operating system of choice. However, PostgreSQL will also run quite well under other versions of Linux, Microsoft Windows, and even Mac OS X.

You can learn more about PostgreSQL at:

http://www.postgresql.org/

### Writing code with Python

The primary programming language of OpenERP is Python. Like the other technologies underlying OpenERP, Python language is open source and will run on all the major contemporary operating systems.

You can learn more about the Python programming language at:

http://python.org/

### Following the Model-View-Controller design

OpenERP is built upon a Model-View-Controller (MVC) architecture. One of the primary goals of this architecture is to separate the visual display of the information from the business rules and management of the underlying data. For example, if you need to change the way data is organized in the model, it is desirable not to have to make dramatic changes to how you view the data. The same is true for maintaining flexibility in viewing data. Today it is common to have many different client applications sharing the same underlying data.
In OpenERP's case, they have often supported a web-based client as well as a desktop client application based on the Gnome Toolkit and known as the GTK client. OpenERP's MVC architecture allows developers to create a wide range of client applications to extend OpenERP.

Starting with Version 7, the GTK client is no longer in development by the core OpenERP team. Instead, GTK development has been handed over to the community.

Designing models
The model is essentially the data that makes up your OpenERP installation, which is stored in the PostgreSQL database. OpenERP is unique in the way that database structures are typically defined by the OpenERP modules at the time they are installed. The OpenERP framework takes the model definitions and automatically creates the necessary table structures inside the PostgreSQL database. Furthermore, a web interface in OpenERP allows administrators to easily extend the OpenERP data model in a variety of ways without having to modify the OpenERP source code.

Rendering views
Each view in OpenERP is defined in XML documents. The OpenERP framework is responsible for rendering these view files in a web browser. Alternative views can be built to render OpenERP functionality upon other platforms, such as mobile devices.

Authoring controllers
The controller component of the architecture is where the business logic and workflow rules of the OpenERP application are applied. The controller components in OpenERP are written in Python code and stored as objects in OpenERP modules.

Choosing your installation operating system
In this section we will discuss some of the advantages and disadvantages of choosing Ubuntu or Windows for your first OpenERP installation.
Choosing a Microsoft Windows OpenERP installation

For the most part, Ubuntu has been the platform of choice for most OpenERP installations. However, there are some reasons why you may choose to run OpenERP under a Windows installation.

Some people who bought this book may have already jumped ahead and installed OpenERP on their Microsoft Windows computer. So, for you go-getters, that working installation of OpenERP may function just fine for researching and testing its features. Often, the Windows all-in-one installer provides a simple method for getting OpenERP up and running in a snap on your hardware. Basically, you do not have to install a new operating system.

If you are familiar with Windows and have no Ubuntu experience, you may get going a little faster by sticking with a Windows installation for your first setup. Downloading and installing modules and making changes to configuration files will be much easier if you are familiar with the operating system.

Choosing an Ubuntu OpenERP installation

While Microsoft Windows does not really need an introduction, it is probably worth giving a brief introduction to Ubuntu. In short, Ubuntu (pronounced ‘oo- boon-too’) is a very popular open source operating system based on the Linux kernel. It has enjoyed increasing popularity because it is easy to install and very stable. Ubuntu can be installed either as a server operating system without a graphical interface, or as a desktop operating system with a graphical interface that closely resembles Windows.

You can learn more about the Ubuntu operating system and why it is so popular at:

http://www.ubuntu.com/

It is generally accepted that Ubuntu is the recommended operating system for running a production installation of OpenERP. There are several reasons why this is true:

Ubuntu is the primary target platform

While OpenERP is released for Windows and is still well-supported, the Ubuntu installation continues to be favored. The development team for OpenERP works primarily with Ubuntu for bug fixes and platform releases. It can be expected that for the most part, OpenERP development will be optimized around the Ubuntu OS, not Windows.
Ubuntu is open source
Installing OpenERP on any Windows operating system is going to require a license from Microsoft. While using OpenERP on your Windows desktop is a viable and perhaps desirable solution for testing and development, it is unlikely you will want to run OpenERP on a Windows desktop system for any production environment. Why? Well, this requires Windows Server, which has much higher license costs than desktop editions. With an Ubuntu installation, you get an entirely open source and virtually cost-free solution.

Ubuntu has additional scalability options
It is possible to configure a more scalable solution under Ubuntu than you can currently configure under Microsoft Windows Server.

Ubuntu has strong community support for OpenERP
The fact is that a vast majority of the production installations of OpenERP are running under Ubuntu. When you run into trouble or management issues with your OpenERP installation, you may find it easier to get assistance if you are running an Ubuntu installation.

At the time of publication, the latest all-in-one installation of OpenERP 7 for Windows requires that you create a login on the OpenERP website to add any applications. However, the same build version in Ubuntu does not require a login. This moderate inconvenience may lead to the Windows versions of OpenERP becoming even less accepted by the active OpenERP community.
Choosing another OS option for OpenERP

Although this book will focus on Windows and Ubuntu installations, you do have several other options. In the past, OpenERP has been deployed under a variety of Linux distributions and even the Macintosh OS. There are also many community members actively developing client frontends for mobile platforms such as Google's Android OS.

Understanding OpenERP releases

When deploying an OpenERP system, it is important to understand the various OpenERP versions as well as the release and upgrade policies. There are two major versions of OpenERP available: the stable version and the trunk version. The stable version is the standard support version of OpenERP and typically the one you should choose to install for most situations. The trunk version is the development version and will often contain bugs and unfinished features. It is primarily downloaded by developers or those wishing to get a look at the latest features.

Upgrading OpenERP

The goal of the OpenERP development team is to release two stable version upgrades each year. OpenERP further labels some stable versions as Long Term Support (LTS) versions. These releases are supported by OpenERP for those that have an OpenERP Enterprise support contract. For any production environment, it is smart to choose an LTS version. Most importantly, installing an LTS release of OpenERP will make bug fixes and patches much easier to implement.

At the time of writing this, the most recent stable LTS version is Version 7.0
Install OpenERP on Windows OS

We begin our installation by locating the packages that are currently available to install. You can find the current list at:

http://nightly.openerp.com/

OpenERP Nightly builds

Every night a new set of packages (exe, tgz, deb, rpm) is generated automatically for the following branches:

- 7.0 LTS (stable)
- 6.1 (unsupported)
- 6.0 LTS (old stable)
- trunk (development)

OpenERP Debian/Ubuntu packages

To install the Debian/Ubuntu package, add the following line to your /etc/apt/sources.list:

deb http://nightly.openerp.com/7.0/nightly/deb/ ./

And type:

sudo apt-get update
sudo apt-get install openerp

The automatic app update mechanism (Settings->Modules->Updates) is currently unavailable on Debian/Ubuntu, however you may use the standard apt-get update, apt-get upgrade dance to keep your openerp installation updated.

OpenERP Nightly builds scripts

Scripts used to build those packages are available at http://openerp-tools.

Building the windows installer requires additional files that can be found at http://windows_manual_build

Older releases 5.0 4.2 4.0 3.2 3.0

Those builds are still available in http://old.releases.
The preceding screenshot is the OpenERP Nightly builds page, that is, the jumping-off point for downloading the source files for installation.

The examples and case studies in this book use OpenERP 7.0. This means you should select the 7.0 LTS (stable) version of OpenERP to download. You can navigate directly to the 7.0 OpenERP downloads here:

http://nightly.openerp.com/7.0/nightly/

Windows installations use the EXE packages. Click on the exe/ directory as shown in the preceding screenshot to get the list of downloads that are available.
Naturally, the specific download packages are going to change on a nightly basis. The latest version of the stable LTS release will contain the most current OpenERP build with bug fixes included and will appear at the bottom of the list. By the way, the upload dates you'll see are in **Coordinated Universal Time (UTC)** and therefore may be many hours ahead of your time zone, especially if you live in the Western Hemisphere.
Performing an all-in-one OpenERP installation in Windows

Installing OpenERP using the all-in-one package is very simple. After the package has finished downloading, double-click on the EXE file to begin the installation wizard.

The first screen will prompt you to select the language for your installation.

After you have selected the language and clicked on the OK button, the wizard will continue with the installation.
After clicking on the **Next** button you will see the license screen followed by a prompt to select the type of install.

I highly recommend that you select the **Custom** install so you can select the directory for installation. The default directory name contains the lengthy build number making it rather difficult to work with in the command prompt.
We have changed the type of install to **Custom** so that we may set the directory name later in the setup wizard.

Next the wizard prompts for the credentials to connect to the **PostgreSQL** database that will be installed.

![Configuration screenshot](attachment:image.png)

It is recommended that you change the username and password for security purposes. The default username and password are shown in the preceding screenshot. These values will be written into the OpenERP configuration file. The username and password provided will be the administration credentials for the **PostgreSQL** database so be sure to remember them.
After clicking on the **Next** button you will be prompted for the installation directory. You will notice that the default installation directory is quite cryptic. While you can use this default directory, it quickly becomes tedious to type when you need to perform operations at the command prompt. Therefore it is recommended that you change the destination folder to something more manageable. For example, why not change the directory name to simply **OpenERP**?
Once you click on the **Install** button, the wizard will finish installing OpenERP to your selected directory.

![OpenERP Installation Wizard](image)

**Testing your Windows installation**

After the wizard is complete, if you leave **Start OpenERP** checked and then click on the **Finish** button, OpenERP should open up in your default browser.

If OpenERP fails to launch, you can look at the *Troubleshooting OpenERP installations* section later in this chapter for solutions to some of the problems commonly encountered during installation.
Setting Up OpenERP

Installing OpenERP on Ubuntu
This book will you walkthrough how to install OpenERP on Ubuntu using the latest all-in-one nightly package. Depending on your Ubuntu installation and how you want to work with OpenERP, there are alternative installation methods.

Modifying sources.list
Installing OpenERP in Ubuntu is easy when you use the Debian repository. You can use any standard text editor, such as Nano, to modify the file /etc/apt/sources.list and add the following line:

```
debug http://nightly.openerp.com/7.0/nightly/deb/ ./
```

Installing the package
After saving sources.list, you can start the installation process by entering these commands into a terminal window:

```
sudo apt-get update
sudo apt-get install openerp
```

The OpenERP packages will be first downloaded and then installed. This is an all-in-one installation and should set up all the necessary packages, PostgreSQL, and library dependencies required to run OpenERP.

Testing your OpenERP installation
Point your browser to http://localhost:8069 and you should see the OpenERP login page appear.

Troubleshooting and OpenERP management tips
Even if you experience no snags when first installing or running OpenERP, it would be wise to peruse the following section to familiar yourself with some of the tricks for dealing with common problems. Many of these guidelines are also useful for managing your system and thwarting potential problems.
Troubleshooting OpenERP installations
As far as ERP installations go, OpenERP is typically very easy to install. Unfortunately, it is possible for an installation to fail for a variety of reasons. In this next section, we will discuss some of the more common installation issues and provide some troubleshooting tips for diagnosing problems with an OpenERP installation.

Checking your browser destination
If you have followed the default installation then your OpenERP installation should be accessing OpenERP at:

http://localhost:8069

Make sure the URL is exactly as shown previously. If you did change the port number during installation, make sure you change the port in the URL.

Verifying that the OpenERP service is running
If you are unable to pull up OpenERP in the browser, it can be good to verify that the OpenERP services are running.

Checking for the OpenERP services running in Windows
Pull up the task manager and navigate to the Services tab, then look for openerp-server-7.0. The status should be Running.
Setting Up OpenERP

The preceding screenshot is an example of the openerp-server-7.0 service successfully running under Windows.

Additional OpenERP troubleshooting steps for Windows can be found here:

https://doc.openerp.com/install/windows/server/complementary_install_information/

Checking for OpenERP services running in Ubuntu

In Ubuntu you can locate the OpenERP services by running the following command in a terminal window:

```
ps aux | grep openerp
```

You will then see the OpenERP service listed if it is running.

Starting and stopping OpenERP services in Ubuntu

When managing an OpenERP server, one of the most common tasks you will find yourself performing is starting and stopping the OpenERP services. OpenERP allows you to start and stop the services with a command switch.

To start the services use the following command:

```
sudo /etc/init.d/openerp-server start
```

To stop the services use the following command:

```
sudo /etc/init.d/openerp-server stop
```

Finding the primary OpenERP logfile

OpenERP writes many messages, warnings, and error messages to a log. Often when troubleshooting problems, this logfile is valuable in determining what action you should take. In a default installation the logfile is located at:

```
{install directory}/server/server/openerp-server.log
```

The log is especially valuable to locate problems you may have while installing new modules.
Modifying the OpenERP configuration file

The OpenERP framework allows you to specify a configuration file for your installation. By default, this file is located at:

/etc/openerp/openerp-server.conf

Using this file, you can change many of the attributes of OpenERP.

Changing port numbers

By default, OpenERP 7 runs on port 8069. For many installations, the default port will work fine. However, there are situations where it can be useful to change this default port. One common scenario would be the need to run more than one version of OpenERP. Multiple installations cannot run on port 8069, so you will need to modify the port. Sometimes there are security reasons behind changing ports as many hackers are aware of the default ports people use.

Fortunately, changing the default port number is easy. Simply specify the following:

Port=[port]

For example, Port=8059 will change the default port for the web client to port 8059.

Changing the admin password

OpenERP offers database management tools that can be accessed easily through your web browser. This makes it easy to create, backup, and even delete a database, all through a web interface. By default, OpenERP sets the password for these operations to admin. To secure your server, it is necessary to change this password in your configuration file:

Admin_password=[your password]

Summary

In this chapter we saw how easy it was to get started using OpenERP online. We discussed how to set up a trial company, the basics for creating a database, and installing your first module. If you chose not to use the online services, you likely found the topics on installing OpenERP on Windows or Ubuntu helpful. Finally, we discussed various methods of troubleshooting and configuring OpenERP.

In the next chapter we will begin to jump into our first real business applications in OpenERP. You will get introduced to our real world case study and set up the basic configuration for the company. We will walk through setting up your first product, and finally creating and printing your first sales order.
Starting Your First Company

We have learned about the various modules that OpenERP has to offer and how you can install OpenERP on your own system. Now we will get into how you can use OpenERP to begin managing your business. We will start by looking at the overall business requirements and decide on the first set of modules we wish to implement. After understanding our basic objectives, we will create an OpenERP database and configure the required company information.

Next, we begin exploring the OpenERP interface for creating and viewing information. We will see just how easy OpenERP is to use by completing an entire sales order workflow. We will finish up the chapter by reviewing some of the more advanced sales order configuration options.

Topics that we cover in this chapter will include:

- Adding a password-protected database to our installation
- Installing and configuring the Sales Management module
- Using interface features to view, edit, and find information
- Entering a new customer
- Adding our first product to sell
- Writing an order and confirming it for invoicing

Gathering requirements

Setting up an OpenERP system is no easy task. Many companies get into trouble believing that they can just install the software and throw in some data. Inevitably, the scope of the project grows and what was supposed to be a simple system ends up being a confusing mess. Fortunately, OpenERP’s modular design will allow you to take a systematic approach to implementing OpenERP for your business.
Implementing an ERP system with a modular approach

The bare bones installation of OpenERP simply provides you a limited messaging system. To manage your OpenERP implementation you want to begin planning which modules you will work with first. OpenERP allows you to install just what you need now and then install the additional OpenERP modules as you better define your requirements. It can be valuable to take this approach when you are considering how you will implement OpenERP for your own business.

Don't try to install all the modules and get everything running all at once. Instead break down the implementation into smaller phases.

Introducing Silkworm – our real-world case study

To best understand how to work with OpenERP, we will build our exercises around a real-world case study. Silkworm is a screen printing, embroidery, and graphic design company. Using OpenERP’s modular design, we will begin by implementing the Sales Order module to set up the selling of basic products. In this specific case we will be selling t-shirts. As we proceed through the book we will continue to expand the system by installing additional modules.

When implementing OpenERP for your organization, you will also want to create a basic requirements document. This information is important for configuration of the company settings in OpenERP and should be considered essential documentation when implementing an ERP system.

Creating a new database in OpenERP

To create your company in OpenERP you must first create a database. As you add additional modules to OpenERP, the necessary tables and fields will be added to this database.
If you have just installed a fresh copy of OpenERP, you will be prompted automatically to create a new OpenERP database.

The preceding screenshot shows the OpenERP form to **Create a New Database**.

OpenERP provides basic instructions for creating your database. Let us quickly review the fields and how they are used:

**Setting our master password**

The master password is set in the OpenERP configuration file. In this form you are not setting the master password but are supplying the master password so that OpenERP can be sure you are authorized to create databases. If you enter an incorrect master password or do not enter a master password, you will get an access denied message when you go to create the database.

By default, the master password for OpenERP is `admin`.

For security reasons, it is essential that you change the default master password.

**Selecting a database name**

When selecting a database name, choose a name that describes the system and will make the purpose of the database clear. There are a few rules. Your database name cannot contain spaces and must start with a number or letter. Also you will need to avoid commas, periods, and quotes. Underscores and hyphens are allowed if they are not the first character in the name.
It can also be a good idea to specify in the name if the database is for development, testing, or production purposes.

For the purposes of our real-world case study, we will use the database name: 

```
SILKWORM-DEV
```

We have chosen the `-DEV` suffix as we will consider this a development database that will not be used for production or even for testing.

Take time to consider what you will name your databases. It can be useful to have standard prefixes or suffixes depending on the purpose of your database. For example, you may use `'-PROD'` for your production database or `'-TEST'` for the database you are using for testing.

### Loading the demonstration data

When selecting a database name, choose a name that describes the system and will make it clear the purpose of the database. There are a few rules. Your database name cannot contain spaces and must start with a number or letter. Also you will need to avoid commas, periods, and quotes. Underscores and hyphens are allowed if they are not the first character in the name.

For the purposes of our real-world case study in this book do not load the demonstration data.

### Specifying our default language

OpenERP offers a variety of language translation features with support for more than 20 languages. All of the examples in this book will use the English (US) language option. Be aware that depending on the language you select in OpenERP you may need to have that language also installed in your base operating system.

### Choosing a password

Each OpenERP database is created with an administrator account named `admin`. This is also known as the superuser account. The password you choose during the creation of the database will be the password for the `admin` account.

Choose any password you wish and click on Create Database to create the `SILKWORM-DEV` database.
Managing databases in OpenERP

The database management interface allows you to perform basic database management tasks such as backing up or restoring a database. Often with OpenERP it is possible to manage your databases without ever having to go directly into the Postgres database server. It is also possible to set up multiple databases under the same installation of OpenERP.

Installing the Sales Management module

After clicking on Create Database, it can take a little time (depending on your system) before you are shown a page to install your first module.

The preceding screenshot shows a screen that lets you select from a list of the most common OpenERP modules to install.
As you can see from the menu on the right, there is very little you can do with just an OpenERP database with no modules installed. Now we will install the **Sales Management** module so that we can begin setting up our t-shirt selling business.

<table>
<thead>
<tr>
<th>Sales Management</th>
<th>Quotations, Sales Orders, Invoicing</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Install" /></td>
<td></td>
</tr>
</tbody>
</table>

Click on the **Install** button to install the **Sales Management** module.

During installation of modules and other long operations, you will often see a **Loading...** message in the center of your screen.

**Configuring accounting data**

With the installation of the **Sales Order** module, OpenERP prompts you to configure the accounting package you will use with your company. For our example, we will be using the **United States - Chart of accounts** option.

The preceding screenshot displays the screen you will receive during the installation of the **Sales Management** module.

**Setting your accounting options**

There are several basic chart templates that are included with OpenERP. These templates include:

- Advertising
- Agriculture
• Construction Trades
• Financial Services
• General Service-Based Business
• Legal Services
• General Product-Based Business

The preceding screenshot displays a screen that is presented during the setup of the Sales Management module.

You are welcome to experiment with the Currency and Sale Tax/Purchase Tax settings. For the purpose of our case study, we have selected the General Product-Based Business chart template and USD currency. Click on Apply to finish installing the Sales Management module.

Knowing the basic OpenERP interface

After the installation of the Sales Order module, OpenERP takes you directly to the Customers form. Let's take a moment to look at the screen elements that will appear consistently throughout OpenERP. In the top-left of the main form you can clearly see that we are in the Customers section.

Using the search box

In the top-right corner of our form we have a search box.
The search box allows you to quickly search for records in the OpenERP application. If you are customers, naturally the search will be looking for customer records. Likewise if you are looking at the product view, the search box will allow you to search the product records you have entered into the system.

**Picking different views**

OpenERP also offers a standard interface to switch between a list view, form view, or other views such as Kanban or graph views. You can see the icon selections under the search box in the right corner of the form.

The currently selected view is shared in dark. If you hover your mouse over the icon, you will get a tooltip that shows you the description of the view. As we have no records in our system currently, let us add a record so that we can further explore the OpenERP interface.

**Creating your first customer**

Helpful instructions prompt you to begin entering your first customer into OpenERP by clicking on the **Create** button.
The preceding screenshot is the Customers form. Clicking on Create will create a customer record.

Silkworm sells t-shirts to both businesses and retail customers. For this example we will use a fictional customer, Mike Smith, who wishes to purchase several t-shirts. OpenERP offers flexibility in collecting customer information, and by default most fields are not required. Three main fields are required in a default installation of OpenERP sales management:

- Customer name
- Accounts receivable account
- Accounts payable account

The rest of the fields are optional. Later we will learn how you can configure OpenERP to make additional fields required.

In this example we have filled out some of the basic fields for our fictional customer Mike Smith:
Is a Company?
At the very top of the form, there is a checkbox to inform OpenERP if this customer is a company. For our example we are using a walk-in retail customer. If you were doing a business-to-business type operation then often your customers would have the Is a Company? checkbox selected.

When you set up a customer as a company, you will have the option to have multiple contacts available for that customer. However, if you leave this option unchecked like our example, you will not have the option to have contacts associated with the customer.

Entering data into an OpenERP form
OpenERP utilizes a consistent interface to enter data throughout the application. Once you have learned how to enter data into one form, you should have no problem entering data into other forms in OpenERP.

Required fields will always be in purple. If you see a purple field, you must fill in that data or you will not be able to save the record. You can move between fields by using your mouse or the Tab key. Shift + Tab will take you back to the previous field. Unlike some systems, you cannot move between fields in OpenERP by using the arrow keys.

In many forms, you will have to select lists that allow you to choose from a list to populate the field. You can use your keyboard to type and limit the items that are displayed in a select list. By using the Tab key and your keyboard to find the appropriate item in the list, it is possible to enter data into a form with limited use of the mouse.

Many select lists have two options at the bottom that will allow you to use additional search options or to create an item that is not in the list.
In this example we see a list of states with the option for additional searching or to create a new state that is not in the list.

Use the Internal Notes area to enter any additional notes that you wish to keep on the customer.

**Editing customer information – the Sales & Purchases page**

The bottom area of the Customers screen is divided into a series of tabs or pages that assist in organizing customer information. In the Sales & Purchases page, we can assign options such as a salesperson and various email options.

The following are the available options in the Sales & Purchases page:
Salesperson
The Salesperson field allows you to select who the direct salesperson will be for this customer. While the field is not required, it is often populated if you are integrating your sales management system with the Customer Relationship Management (CRM) module. We will use this field in Chapter 3, Exploring Customer Relationship Management in OpenERP; for now we can leave the field blank.

Reference
Often when implementing OpenERP, a company already has an existing customer numbering system in place. The Reference field is the perfect field to populate with an existing customer number. Otherwise this field can be left blank or used for another purpose. For our example we are going to leave this field blank.

Language
OpenERP has the ability to work with customers in a variety of languages. For our example we will select English. If however you were working with a company that preferred their documents in other languages, you could specify that language and OpenERP will manage the translation.

Date
The Date field does not specify exactly what date this refers to for the customer. In most implementations, the business would define this date to be the date the customer was acquired. Depending on your needs, you could define the customer date to have an alternative definition. It is also perfectly acceptable to leave this field blank, as we will in our example.

Customer
The Customer checkbox is known in OpenERP as a Boolean field. It is either yes or no, on or off. OpenERP has a unique method of storing data related to people in the system. All individuals are stored in the same table (res_partner) regardless of whether they are a customer or supplier. The Customer flag tells OpenERP that this is in fact a customer record. This field must be checked for OpenERP to recognize Mike Smith as a customer.

Supplier
Because OpenERP stores customer and supplier data in the same table, it is possible to be both a customer and a supplier. In this example we will keep Mike Smith as a customer only.
OpenERP uses a common database to store customer and supplier records. This makes it easier to manage data and customers information, and suppliers are designated by simple checkboxes in the Sales & Purchases page on the Customers screen.

**Active**

The Active flag allows you to remove a customer from the customer list without deleting them from the database. An example use for the Active field would be to uncheck it if a customer has not made a purchase in few years. For our example, we will naturally leave this checked.

OpenERP provides an Active field for most records in the system. This allows you to easily make a record inactive without having to remove the record from the database.

**Receive Messages by Email**

The Receive Messages by Email option allows you to decide the communication level you wish to have with your customer and under what conditions they should receive e-mail. The available options include:

- Never
- Incoming Emails only
- Incoming Emails and Discussions
- All Messages (discussions, emails, followed system notifications)

**Opt-Out**

The Opt-Out flag will allow you to prevent any automatic or campaign e-mails from being sent to the e-mail address in the customer record. For the purposes of our example we will leave the default settings.

**Editing customer information – the Accounting page**

At first the Accounting page on the Customers screen can feel a bit intimidating but to enter a new customer we must provide some essential information. Fortunately there are only two required fields on this page: the Account Payable and Account Receivable fields.
We will leave the more complex accounting configuration to a later chapter.

The following are the available options in the Accounting page:

**Fiscal Position**

The Fiscal Position field is sometimes also known as the tax status and in some systems is represented simply as taxable. In OpenERP you have two options for Fiscal Position:

- Normal Taxes
- Tax Exempt

Tax exempt is common in business-to-business situations in which taxes are waived because the customer is purchasing the product for resale. This field is not required, and it is possible to override this selection when producing a sales order.

**Account Receivable**

The Account Receivable field specifies the default accounts receivable account for the customer. This field is required, and the account will be automatically debited when a customer is invoiced. When the invoice is paid, then the accounts receivable account will be credited.

**Customer Payment Term**

It is common in many businesses for different customers to have different payment terms. Perhaps for a lifelong customer you would extend 30 or even 60 day net terms to pay their invoice. For a new customer you may require immediate payment. Additional terms can be configured in OpenERP depending on your needs.
The default payment terms included are:

- Immediate Payment
- 15 Days
- 30 Net Days

For our example we will set the payment term to **Immediate Payment**.

**Total Receivable**

The **Total Receivable** field is a computed field and is currently $0.00 because this is a new customer. As customers are invoiced this field will change to reflect how much they currently owe.

**Credit Limit**

The **Credit Limit** field allows you to establish credit limits for your customer. The system can then configure warnings to alert you if a sales order would push a customer beyond their credit limit. For our example we have **Immediate Payment** required so we will leave the credit limit at $0.00.

**Latest Reconciliation Date**

**Latest Reconciliation Date** is the date that the accounting entries for the customer were last reconciled. Because there have been no automatic or reconciliation operations performed, this field is blank.

**Account Payable**

While this account is required, most likely it will never be utilized by customers in OpenERP. However if the **Supplier** field is checked, then this would be the accounts payable account used in supplier-related transactions. Still, you will need to specify an accounts payable account to finish entering the customer.

**Supplier Payment Term**

Much like the customer payment term, this field will determine the payment terms for the supplier. Because a partner can be both a customer and supplier we have separate terms for each.
Bank Accounts
At the very bottom of our Accounting screen, we can set up optional bank accounts for our customer. Clicking on Add an item will bring up a Create: Banks screen to collect information that would be valuable in sending payment data or issuing automated clearing house (ACH) drafts against a customer's bank account. For our example we will not enter a bank account.

Saving the customer record
With the basic customer information entered, we can now click on the Save button to commit our changes to the record.

Entering a product in OpenERP
Now that we have a customer, it is time we enter some products to sell to our new customer. For our example we are going to enter a medium white cotton t-shirt. Click on the Products item in the menu on the left as shown in the following screenshot:
You will notice that even without installing demonstration data, OpenERP still creates a default product named **Service** with a price of $75.00. We will leave this product in the system and create a new product by clicking on the **Create** button.

---

**Product Name**

The product name is what will display on the sales orders, invoices, and in all other screens which refer to this specific product. For our example we are selling a **Medium White T-Shirt**.

**Category**

By default, products will be added to the **All products** category. Depending on the needs of your business, you can define additional categories that best represent and organize your product line. It is possible to set up subcategories and even sub-subcategories if we have a complex mix of products. For our example we will leave the default **All products** category.
Can be Sold
Much like the Active flag on the Customers screen, you can use the Can be Sold option to prevent products from showing up on the product list by unchecking Can be Sold. For our example we want to sell this t-shirt to Mike Smith so we will leave the option checked.

Entering a product – the Information page
The following are the available options in the Information page:

Product Type
Product Type is the first option in the Information page on the Products screen. There are two available product types:

- Consumable
- Service

The Service product type will not create procurements in purchase orders. The Consumable product type is a product that you actually sell and can be configured to generate purchase orders. For our example we will set the product type to Consumable.

Sale Price
The Sale Price field sets the sales price of the item as it will appear on the sales order. For our example we are setting the sales price of the t-shirt to 16.50.

Internal Reference
For the most part, OpenERP utilizes the Name field and the Description field when displaying the product information. It is very common that a company may have a coding system for their products. The Internal Reference field is useful to enter an alternative product code or number for the product. In this example we will leave the Internal Reference field blank.

EAN13 Barcode
OpenERP provides the EAN13 Barcode field so that product records can be easily integrated with the scanning solutions. For now we will be leaving this field blank.
Entering a product – the Procurements page

The second page on the Products screen collects information related to procurements.

The Procurements page on the Products screen

The following are the available options in the Procurements page:

**Cost**

Even without the purchasing module installed, it is possible to specify a cost for the product. This number will be used for standard stock valuation in accounting and will also serve as the base price on purchase orders once the purchasing module is installed. For our example we are just going to leave the Cost field set to 0.00.

**Description for Suppliers**

It is not uncommon in business to use different names for the same products depending on whether you are talking to a supplier or a customer. This field allows you to specify the product description for the supplier. Some industries will find this essential for communicating product information to the supplier. For our example we will leave this field blank.

Entering a product – the Inventory page

The Inventory information page lets you collect information on the current status of the product and assign a product manager.

The Status page from the Product form

The following are the available options in the Inventory page:
Status
The Status field allows you specify the various product stages and provides an additional level of classification in the inventory. The default status values are:

- In Development
- Normal
- End of Lifecycle
- Obsolete

For our example we will select the Normal status. This is not a required field and the product can be still entered on a sales order if this field is left blank.

Product Manager
Each product can be assigned a dedicated product manager. This can be useful for reporting purposes and creating intelligent dashboards. For example, an engineer could have their dashboard configured to only show them the products in which they are the product manager. For our example we will leave this field blank.

Entering a product – the Sales page
The Sales page on the Products form allows you to specify optional information on the product as it relates to sales and quotations. If there is a warranty on the product, you can specify the warranty duration in months. You can also specify a description that will show up just on quotations. This would be used, for example, if you want a different description on the quotation than you have on the invoice. We will leave these fields as defaults for our case study.
Entering a product – the Accounting page

We set up a default tax of 9.50% when we installed the Sales Order module. However there will be times when you have a product that has a specific tax. One example is that cigarettes often have more tax than other items. OpenERP allows you to specify additional tax options for a given product in the Accounting page. Taxes can be specified for both the customer and the supplier.

For our example we will set the Customer Taxes field to 9.50% — the rate we defined in the sales order accounting setup.

Saving the product

Clicking on the Save button saves the product record in OpenERP. If you click on Discard, you will get prompted with a warning message that you will lose your changes.

Setting the company information

We have entered both a customer and a product. However before we create a sales order, we still have some work to do in setting up our company. Currently OpenERP does not even know the name of our company and has by default used Your Company as the name.
Starting Your First Company

We can locate the company's information by choosing Settings from the top menu and then choosing Companies from the submenu on the left.

List of the companies in the current OpenERP database

You can now left-click on Your Company to open up the company information screen. Click on the Edit button so we can enter the edit mode and update the company information.

A company record filled in with the data for our sample case study
Here we have supplied the company name along with address, e-mail, phone number, website, and the company slogan. It is also possible to click on the photo icon at the top-left corner of the screen to assign a logo to the company. At the bottom of the screen, you can add bank accounts for the company. We will wait to configure the bank accounts for a later chapter.

The Configuration page
The Configuration page simply allows you to set the default currency for the company. This field was set to USD (US Dollars) during the Sales Management module set up.

The Overdue Payments page
The Overdue Payments page will allow you to change the statement that will appear to customers who are flagged for being late with their payments. For our example we will keep the default.

Saving company information
Click on Save to save the company information. We are now ready to enter our first sales order.

Entering your first sales order
Now is the moment we have all been waiting for. We finally get to sell our products by entering a sales order. To get to the sales order entry, click on Sales in the top menu and then choose Sales Orders from the submenu on the left.
The preceding screenshot lists existing sales orders and allows users to create a new sales order.

Click on the **Create** button to create a new sales order.

![New Sales Orders form with the cursor set on the Customer field](image)

**Selecting the customer**

When you create a new sales order, you are prompted to first select the customer from the select list. As you add customers you will have the option to search and locate customers for the sales order. For now we will be selecting the customer we entered in early in the chapter: **Mike Smith**.
You will not be able to begin entering line items until you have specified the customer for the sales order.

**Date**
By default the current date is populated into the **Date** field. If necessary you can modify this date. Clicking on the little calendar will pull up the date selection field. For our example you can just use today's date.

![Selecting a date in OpenERP](image)

**Customer Reference**
The **Customer Reference** field can be used to collect information that you may wish to associate with the sale. For example you may wish to store a reference about how the customer was acquired. We are leaving this field blank for our example.
Entering line items on a sales order

Now we are ready to begin specifying the product we wish to sell. Click on Add an item in the line item area to add a line to the grid. The first field will be Product. Select Medium White T-Shirt from the listbox. Your line item fields should populate and look like the following screenshot:

![Quotation /](image)

**Product**

Each line item starts out by selecting the product. You can add products on the fly by choosing Create and Edit... from the bottom of the list. Once there are more products in the list, you can also bring up a Search: Product window using the Search More... option. After you select the product field, OpenERP retrieves the tax and pricing information from the server to display in the line item.
OpenERP does not automatically refresh **Total**. If you wish to see the total before clicking on the **Save** button, click on the **(update)** link located next to **Total** at the bottom of the form.

**Description**
OpenERP will pull over the description from the product record to populate the **Description** field on the line item. It is possible to override the description on the sales order. For this example we will leave the description as it was pulled over from the product record.

**Quantity**
The product quantity will default to one. Naturally, you will change this field to the quantity of products you have sold. We will just leave quantity as one for this example.

**Taxes**
OpenERP supports taxes by line item and will automatically pull over the 9.50% tax rate that we have defined in the product record. Additional taxes can be added or removed from the line item. For this example we will leave the tax at 9.50%.

**Unit Price**
OpenERP pulls the sale price from the product record to populate the unit price in the line item. It is possible to override the price in the line item. For this example we will leave the unit price at 16.50.

Be careful when changing prices in the line items of OpenERP. It is possible that if you click back on the **Product** field or tab through other fields in the line item that the unit price will flip back the price in the product record. If you are changing prices in the line items, make sure to double check your unit prices before you confirm your sales order.
Saving the quotation
Click on Save to save the quotation. The form will refresh, displaying the full customer address as well as updating the tax and final total of the sales order.

Understanding the sales order workflow
Although we started out entering a Sales Order, the current state of this order is a Draft Quotation. OpenERP 7 displays the current state of transactions in the top-right corner of the form.
This indicator makes it very easy to see at what stage a transaction is, throughout the OpenERP workflow. In this example, we can see that the sales order is currently in the Draft Quotation stage. We can also see that the quotation would first need to be sent before the quotation can finally be considered Done.

The available actions you can take on this quotation are displayed at the top-left corner of the form.

---

**Send by Email**
Clicking on the **Send by Email** button will send a copy of the quotation to the e-mail address in the customer's file. Setting up your e-mail configuration will be a topic for another chapter.

**Print**
Even in the digital age it is still very common to need a printed copy of a quotation or sales order. Clicking on the **Print** button will generate a PDF document containing your quotation.

**Confirm Sale**
The **Confirm Sale** button will convert the quotation into a sales order and push the transaction further down the sales workflow.

**Cancel Quotation**
Clicking on the **Cancel Quotation** button will prompt you to cancel this quotation. The quotation is not deleted and can still be viewed. Cancelling the quotation ends the sales order workflow and the quotation will only be kept in the system for archive purposes.

Click on **Confirm Sale** to convert this quotation into a sales order.
Invoicing the sale
Depending on the workflow of the business, a lot of different things can happen after you have confirmed a sales order. In manufacturing companies, you may need to purchase products as well as create a manufacturing order to produce the final product before you invoice the customer. In our example, we are going to go ahead and invoice the customer for the t-shirt they have ordered. Click on the Create Invoice button to generate an invoice for the sales order.

An OpenERP Invoice Order wizard pops up to walk you through the invoice creation process.

The Invoice Order wizard

What do you want to invoice?
OpenERP provides a variety of options for invoicing the entire sales order or instead invoicing based on other methods. The available choices are:

- **Invoice the whole sales order**
- **Percentage**: If selected, you are prompted to specify what percent of the total sale you wish to invoice in advance.
- **Fixed Price (deposit)**: Choose this option if you have received a deposit on a product. You will be prompted to enter the amount of the deposit as well as the opportunity to designate an advance product.
- **Some Order Lines**: This option will prompt you to select specific line items you wish to invoice.

Creating the invoice
For our example we will be invoicing the whole sales order. Click on Create Invoice to generate the invoice. Initially the invoice is created in a Draft state. Clicking on Validate will confirm the invoice and post the transaction.
If you have followed along and everything worked as it should, then you will see an invoice similar to the invoice shown in the following screenshot:

![Invoice Screenshot]

**Summary**

In this chapter we started by creating an OpenERP database. We then installed the Sales Management module and created our first customer. With our customer created, we turned our attention to setting up a product in OpenERP and entering our basic company information. Next, we created a quotation and followed the workflow all the way through to confirming the sales order and generating an invoice.

In the next chapter, we will look at the Customer Relationship Management (CRM) module and how we can utilize the communication features in OpenERP. There is also a good place to define our sales staff hierarchy. We will learn to create an opportunity, schedule calls, log interactions, and turn our opportunity into a sales quotation. Converting leads into customers will also be covered. Before we tackle all of that, let's look at our sales strategy and what we want to achieve via the CRM software.
Exploring Customer Relationship Management in OpenERP

Until recently, most ERP systems had very product-focused designs. While records and fields maintained basic customer information, processes and reporting typically revolved around product-related transactions. While the business was centered around products before, the shift is now to center the business around the customer. Customer Relationship Management systems, or CRM in short, provide the tools and reporting necessary to manage customer information and interactions.

The topics we cover include the following:

- Developing a plan for implementing our CRM system
- Installation and configuration of the CRM module
- Defining our sales staff and assigning them to customers
- Managing new leads and sales opportunities

Using CRM as a business strategy

Before jumping into the specific CRM features of OpenERP, it is valuable to briefly discuss the importance of a comprehensive approach to implementing CRM in your business. The fact is that successfully implementing a CRM system requires much more planning than just installing software and asking employees to fill in data. CRM software systems are only a technical tool in assisting your sales and marketing department in acquiring and keeping customers.
Certainly, the software will play an important role, but to obtain real benefits from a CRM system, you must do hard research to understand your customer and exactly how you wish to shape up their customer experience.

It is critical that sales people share account knowledge and that they completely understand the features and capabilities of the system. Often, sales people have existing tools that they have relied on over many years. Without clear objectives and goals for the entire sales team, it is likely they will not use the tool. Plan to spend time training and encouraging the sharing of knowledge to successfully implement a CRM system.

Managing the customer experience
Today, customers face a wide range of choices when it comes to purchasing products and services. At a most fundamental level, customers often build great loyalty to brands in which they have a positive customer experience. Companies such as Apple and Harley Davidson are successful largely because of fierce brand loyalty based on positive customer experiences. Making the most of a CRM system requires you to put yourself in the role of your customer and developing a consistent strategy to improving their overall customer experience.

Treating your customer like a real person
As computers became more common, it wasn't long until people began to feel as if they were being treated as numbers by many companies. In many ways CRM systems turn the tables around. Instead of treating customers like cattle, a smart account manager using a CRM system can greatly personalize customer experience. You treat your customer like an individual and they will reward you with their loyalty.

Because you are looking to create a very personalized customer experience, it is important to thoroughly look at your customer's interactions with the company while designing your own CRM system. A company that sells high-end security systems to government institutions will need to provide customers with drastically different customer experiences than a company that is marketing a pool maintenance service.
Using your mission statements and company goals to drive the design of your CRM system

A good CRM system will build around the core goals and mission of your company. If your company does not have customer-focused goals or a mission statement, you should address that before beginning to design a CRM system. Most critically, there needs to be focus on concerns and interactions that have a direct impact on customer experience. A good CRM system will not just manage the sales process but the entire customer experience and interactions before and after the sale.

Real-world case study – improving customer experience

Now, we will take a detailed real-world look at how a CRM system can be implemented to improve customer experience. We begin by looking at the company slogan:

"We make great first impressions last."

Here we have a slogan that most certainly speaks to the value of customer experience. To make that great first impression and keep it, there are several critical service expectations as follows:

- Orders must be accurate and easy for customers to place
- Orders must be delivered on time
- The quality must be excellent

While listing these customer service goals may seem obvious, explicitly naming your objectives is important while building a CRM system. There is a natural tendency while building a CRM system to focus almost exclusively on customer acquisition and presale activities. We must take care to remember that a CRM system must also support processes that manage the entire customer experience. How are problem orders handled? How is the customer contacted if there is a product back order? If the customer calls, can the service representative easily provide delivery tracking information? These are the kinds of scenarios you will want to consider while building your own CRM system.
Exploring Customer Relationship Management in OpenERP

Installing the CRM application
If you have not installed the CRM module, log in as the administrator and then click on the Settings menu. In a few seconds the list of available apps will appear. The CRM will likely be in the top-left corner.

Click on Install just below CRM to set up the CRM application.

Assigning the sales representative or account manager
In OpenERP, like in most CRM systems, the sales repetitive or account manager plays an important role. Typically, this is the person that will ultimately be responsible for the customer account and a satisfactory customer experience.

While most often a company will use real people as their salespeople, it is certainly possible to instead have a salesperson record refer to a group or even a subcontracted support service.
We will begin by creating a salesperson that will handle standard customer accounts. Note that a sales representative is also a user in the OpenERP system.

Create a new salesperson by going to the Settings menu, selecting Users, and then clicking on the Create button. The new user form will appear. We have filled in the form with values for a fictional salesperson, Mike Zeigler.

![The form for entering a new user into OpenERP](image)

**Designating language and time zone**

OpenERP allows you to select the language for each user. Currently, OpenERP supports more than 20 language translations. Specifying your time zone in the Timezone field allows OpenERP to coordinate the display of date and time on messages.

Leaving Timezone blank for a user will sometimes lead to an unpredictable behavior in the OpenERP software. Make sure you specify a time zone while creating a user record. Better yet, check the customization chapter on how you can make time zone a required field!
Receiving messages via e-mail

In OpenERP 7, messaging became a central component of the OpenERP system. Therefore, determining the appropriate handling of e-mails and the circumstances in which a user will receive e-mails is very important. The **Receive messages by email** option lets you determine when you will receive e-mail messages from notifications that come to your OpenERP inbox.

For our example, we have chosen to never receive e-mails. This is the setting you should also use if you have still not configured your e-mail server in OpenERP. Let's review the user options that will be available for communication via e-mail.

- **Never**: Selecting *Never* suppresses all e-mail messaging for the user. Naturally, this is the setting you will wish to use if you do not have an e-mail server configured. This is also a useful option for users that simply want to use the built-in inbox inside OpenERP to retrieve their messages.

- **Incoming Emails only**: Selecting this option will direct OpenERP to send you an e-mail notification whenever an e-mail comes into your OpenERP inbox.

- **Incoming Emails and Discussions**: This option sends an e-mail notification when an e-mail comes in to your inbox as well as any time there is a discussion on a transaction or form that you are following.

- **All Messages (discussions, emails, followed system notifications)**: This option sends an e-mail notification for any action that would create an entry in your OpenERP inbox. Unlike the other options, this action could include system notifications or other automated communications.

**Email**

In this field you specify an e-mail address for the user. The **Email** field is important for operations such as notifications and resetting your password. There are also some operations in OpenERP that simply will not function if you have not provided a valid e-mail address.

In a standard OpenERP 7 installation, you will need to leave email blank until you have properly configured an e-mail server. E-mail server configuration is addressed in Chapter 10, *Customizing OpenERP for Your Business*.
Default Sales Team
Even if you do not load demo data during an OpenERP installation, the system will have one sales team record named Sales Department. Sales teams allow you to organize salespeople in OpenERP. For example, you could have sales teams that are organized by region or by product category.

Signature
The signature section allows you to customize the signature that will automatically be appended to OpenERP-generated messages and e-mails.

Access Rights
The Access Rights tab lets you control which applications the user will be able to access. Mr Zeigler is a sales manager; therefore, we will give him access rights for viewing all the leads. There is no need, however, to give Mr Ziegler access to Accounting & Finance or any administration privileges.
Selecting sales application settings

While setting up your sales people in OpenERP, you have three different options on how much access an individual user has to the sales system.

- **User: Own Leads Only**: This is the most restrictive access to the sales application. A user with this access level is only allowed to see the leads they have entered themselves or which have been assigned to them. They will not be able to see leads assigned to other salespeople in the system.

- **User: All Leads**: With this setting, the user will have access to all leads within the system.

- **Manager**: This setting is the highest access level in the OpenERP sales system. With this access level, the user can see all leads as well as access the configuration options of the sales application. This setting also allows the user to access statistical reports.

Manually setting the user password

You may have noticed there is no visible password field in the user record. That is because the default method is to e-mail the user an account verification they can use to set their password. However, if you do not have an e-mail server configured, there is an alternative method for setting the user password.

After saving the user record, use the More menu at the top of the form and select **Change Password**.

A form will then appear allowing you to set the password for the user.
Assigning a salesperson to a customer

Now that we have set up our salesperson, it is time to assign him his first customer. Previously, no salesperson had been assigned to our one and only customer, Mike Smith. So let's go to the sales menu, click on Mike Smith to pull up his customer record, and assign him Mr Ziegler as his salesperson.

The preceding screenshot is of the customer screen opened to assign a salesperson. Here, we have set Salesperson to Mike Zeigler and Sales Team to Sales Department. By assigning your customers a salesperson, you can then better organize your customers for reports and additional statistical analysis.
Exploring Customer Relationship Management in OpenERP

Tracking leads and opportunities
OpenERP provides two primary documents for managing interactions with your customers or potential customers. You can think of leads as less critical and perhaps less likely to turn into a real sales situation than an opportunity. A good example of leads would be that you get a few dozen business cards from people you met at a conference. You could add each of them as a lead for further follow up. An example of an opportunity would be if you met someone at the conference and had a detailed conversation on how your company provides appropriate services.

Creating a new opportunity
In OpenERP, a potential sale is defined by creating a new opportunity. An opportunity allows you to begin collecting information about the scope and potential outcomes for a sale. These opportunities can be created from new leads or an opportunity can originate from an existing customer. For our real-world example, let’s assume Mike Smith has called and was so happy with his first order that he now wants to discuss using Silkworm for his local Lil League team. After a short conversation, we decide to create an opportunity by clicking on the Opportunities button and then creating a new opportunity.
Subject
The subject used for your opportunity can be anything you wish. It is naturally important to choose a subject that makes it easy to identify the opportunity in a list. This is the only field required to create an opportunity in OpenERP.

Expected revenue and percentage
In the Expected Revenue field you specify the amount of revenue you can expect from the opportunity if you are successful and then the percentage likelihood that this opportunity will result in a sale. These values are useful in many statistical reports, although they are not required to create an opportunity.

Customer
This field is automatically populated if you create an opportunity from the customer form. You can, however, assign a different customer if you like. This is not a required field, so if you have an opportunity that you do not wish to associate with a customer, that is perfectly fine. For example, you may leave this field blank if you are attending a trade show and expect to have revenue, but do not yet have any specific customers to attribute to the opportunity.

Next Action
While following up on your opportunities, one of the most important triggers will be the next action date. Here, you decide when you should next take some sort of action on the opportunity. You are provided with a small note field to remind you of the action that you should be taking. This could be anything from placing a phone call to sending an e-mail to performing a presentation. For our example, we intend to present a proposal on the action date.

Expected Closing and Priority
While managing your opportunities, it is important to establish a goal for when you wish to close the sale. Providing an expected closing date is handy for managing opportunities and running reports identifying which opportunities are due to be closed. The priority setting ranges from lowest to highest, with three settings in between. While defining your CRM system, you should identify business rules for determining under what conditions an opportunity will receive the highest priority.

Categories
OpenERP allows you to assign multiple categories to an opportunity. For example, you could choose Trade Show and Sports as categories to designate an opportunity that is sports-related and will take place at a trade show.
Email and Phone
The Email and Phone fields allow you to specify the primary contact methods you will likely use to communicate with your opportunity.

Internal Notes
The Internal Notes area is where you provide all the details on the opportunity. For our example, we kept the notes brief, but when you are working with real opportunities, make sure that you take advantage of this area to document anything that will help you in closing the sale.

The Lead tab
When you create an opportunity from either a customer or a lead, the information is automatically brought over into the Lead tab in the opportunity.

Entering lead address and contact information
The top half of the Lead tab contains the standard address and other contact information. This information is automatically populated, but it can be overwritten for the opportunity if you desire. The Function field could be used to provide a bit of detail on the event that triggered the opportunity.
OpenERP does not provide separate fields for first and last names like many other ERP/accounting systems. Consider this as you plan how to organize customers in your system.

Mailings
The Opt-Out checkbox under Mailings prevents the lead or customer associated with this opportunity from receiving mass mailings.

Active
This field under Misc is useful if you have an opportunity that perhaps has gone cold. Instead of deleting the opportunity, you can make it inactive by unchecking the box. Later, if the opportunity becomes viable again, you can make it active once again.

Referred By
This is a simple text field under Misc that is not tied to any other data. It is just a place where you can make a note of who may have referred this opportunity to you.

References
The two places to specify references at the bottom of the screen have a great deal of flexibility in tying other information in OpenERP to the opportunity. Not only can you select the type of reference, but you may also tie the opportunity directly to many records in the system. The list includes the following:

- Partner
- Product
- Invoice
- Voucher
- Sales Order
- Event
- Meeting

Save the opportunity by clicking on the Save button at the top of the form.
Looking at your opportunities in kanban view

When you navigate to the Sales menu and choose Opportunities, you will see your opportunities displayed in the kanban view. Here we see our brand new $2,500 opportunity along with the customer and the next action we need to take.

Clicking on the small arrow on the kanban card will bring up a small menu allowing you to perform actions related to the opportunity.

Taking action on an opportunity

Selecting the Edit… option takes you to the opportunity record and into edit mode for you to change any information. In addition, you can delete the record, send an e-mail to the contact associated with the opportunity, and schedule a call or meeting. The color palette at the bottom lets you color code your opportunities in the kanban view. The small star on the opportunity card allows you to highlight opportunities for special consideration. You can also easily drag-and-drop the opportunity into other columns as you work through the various stages of the sale.
Using OpenERP's OpenChatter feature

One of the biggest enhancements in OpenERP 7 is the new OpenChatter feature that provides social networking style communication to business documents and transactions. As we work on our brand new opportunity, we will utilize the OpenChatter feature to demonstrate how to communicate details between team members and generate log entries to document our progress.

The best thing about the OpenChatter feature is that it is available for nearly all business documents in OpenERP. It also allows you to see a running set of logs of the transactions or operations that have affected the document. For our example, let us assume that we have prepared our proposal and made the presentation.

It is time to update the status of our opportunity by clicking on the Proposition arrow at the top of the form.

At the bottom of the opportunity, you will now see a logged note generated by OpenERP that documents the changing of the opportunity from a new opportunity to a proposition.

Managing the opportunity

With the proposal presented, let us take down some details from what we have learned that may help us later when we come back to this opportunity. One method of collecting this information could be to add the details to the Internal Notes field in the opportunity form. There is value, however, in using the OpenChatter feature in OpenERP to document our new details.
Most importantly, using OpenChatter to log notes gives you a running transcript with date and timestamps automatically generated. With the generic notes field, it can be very difficult to manage multiple entries. Another major advantage is that the OpenChatter feature can automatically send messages to team members’ inboxes, updating them on the progress. Let us see it in action!

Click on the Log a note button to attach a note to our opportunity.

When you create a note, it is attached to the business document but no message will be sent to followers. You can even attach a document to the note by using the Attach a File feature. After clicking on the Log a note button, the note is saved and becomes part of the OpenChatter log for that document.

**Following a business document**

OpenERP brings social networking concepts into your business communication. Fundamental to this implementation is that you can get automatic updates on a business document by following the document. Then, whenever there is a note, action, or a message created that is related to a document you follow, you will receive a message in your OpenERP inbox. At the bottom right-hand corner of the form, you are presented with the options for when you are notified and for adding or removing followers from the document.
In this case, we can see that both Mike Zeigler and Administrator are set as followers for this opportunity. The Following button at the top indicates that I am following this document. The checkbox next to Discussions indicates that I should be notified of any discussions related to this document. I would not, however, be notified, for example, if the stage changed. Using the Add others link, you can add additional users to follow the document.

**Scheduling a call**

Now that we have our proposal out and in the hands of our customer, we should schedule a call to follow up. Clicking on Schedule/Log Call brings up a form. If you have already made the call and just wish to log it, you can choose Log a Call. For our purpose, we need to choose Schedule a Call so that we can specify the date and time. Most of the information of the form is filled out automatically. We can set the date and time and then provide some notes for the purpose of our call.

When you click on the Schedule Call button, the call is scheduled in your calendar and a log entry is placed in the opportunity document.
Looking at your scheduled calls

Naturally, there will come a point in which you are looking through the actions that you need to perform for the day. Clicking on the Sales menu and then choosing Scheduled Calls will bring up a list of the scheduled calls you need to make. Here, you will see the call that we scheduled to discuss the Lil League Jerseys proposal.

Clicking on the little green arrow on the far right will mark the call as held. You can also click on the record to bring up a screen to indicate that the call was placed.

Negotiating the sale

Thanks to our CRM system and a timely follow-up phone call, we are poised to close this sale. We now are more confident that we are going to close the sale and are ready to enter the negotiation phase of the sale. Let us document the new developments in our opportunity record.

Open up the opportunity and edit the record to show the expected revenue of $3750.00 and percentage of 80%. We also want to set the stage to negotiation.
Notice at the bottom in the OpenChatter feature that we can see a log of the changes in the expected revenue as well as the upgrade to the stage from Proposition to Negotiation. You should now begin to see the value of the running log of notes and changes associated with documents inside OpenERP.

**Modifying the stages of the sale**

We have seen that OpenERP provides a default set of sales stages. Many times, however, you will want to customize the stages to best deliver an outstanding customer experience. Moving an opportunity through stages should trigger actions that create a relationship with the customer and demonstrate your understanding of their needs. A customer in the qualification stage of a sale will have much different needs and much different expectations than a customer that is in the negotiation phase.

For our case study, there are sometimes printing jobs that are technically complex to accomplish. With different jerseys for a variety of teams, the final details need to go through a final technical review and approval process before the order can be entered and verified.
From a business perspective, the goal is not just to document the stage of the sales cycle; the primary goal is to use this information to drive customer interactions and improve the overall customer experience.

To add a stage to the sales process, switch to the kanban view and then click on the **Add a new column** link in the top left of the form.

You can find the kanban button in the top-right corner of the form.

![Add a new column](image)

We are required to fill in the **Stage Name**, **Related Status**, **Probability (%)**, and **Type** fields for our new stage. The **Related Status** field denotes which predefined status will be assigned to the case when it reaches this stage. The **Probability** field is a prediction of the average success rate for a case coming as far as this stage. The **Type** field tells the system whether to make our new stage available for leads, opportunities, or both of these.

![Edit column](image)

After you have added the column to the sales process, you can use your mouse to drag-and-drop the columns in the order you wish them to appear. We are now ready to begin the **Technical Approval** stage for this opportunity.
It can be confusing what the difference is between status and stage. Think of status as the bigger picture and stages as the smaller, more detailed steps required to get there. For example, an opportunity could be set to the pending status for a variety of reasons.

Drag-and-drop the **Lil League Jerseys** opportunity over to the **Technical Approval** column in the kanban view.

![Kanban view with Lil League Jerseys opportunity moved to Technical Approval column](image)

We now see the Technical Approval column in our kanban view and have moved over the opportunity. You will also notice that any time you change the stage of an opportunity that there will be an entry created in the OpenChatter section at the bottom of the form. In addition to the ability to drag-and-drop an opportunity into a new stage, you can also change the stage of an opportunity by going into the form view.

**Closing the sale**

After a lot of hard work we have finally won the opportunity, and it is time to turn this opportunity into a quotation.

Click on the **Make Quotation** button at the top of the opportunity form.

Taking this action will bring you to the **Make Quotation** screen where you can confirm the customer and mark the opportunity as **Won**.
Clicking on the Create button moves our opportunity to the Won stage and automatically populates the reference field in the opportunity with a link to our newly created quotation.

**Converting our opportunity into a quotation**

The workflow in OpenERP handles moving over all the required information from our opportunity to our quotation document. Our original reference to the opportunity is located in the Source Document field in the Other Information tab. At this point, we are ready to begin adding line items and creating a quotation just like we did in Chapter 2, Starting Your First Company.

**Creating leads in OpenERP**

Many times it can take quite a bit of work to uncover an opportunity. In OpenERP, we create leads when we need a qualification step before creating an opportunity or a customer. For example, we may receive a business card or an unqualified lead from our website. Let’s create a new lead for a potential customer we met at a local event.

Under the Sales menu, click on Leads and then the Create button to open a new lead.
As you will see, the form is very similar to the standard customer screen. In our example, we have filled out the basic contact and address information as well as assigned our sales representative to this lead.

For this example, we are not yet creating a customer. Notice, however, that there is a customer field available in the form. It is possible we could come across a lead that perhaps is tied to an existing customer we already have in the system. In this case, we could select the customer and all the rest of the information would be filled in.

Typically, if the lead is not qualified, we would leave the customer field blank. On the other hand if this was a lead in which we already feel we could create and send them a quote, we could create a customer directly from this form.

After you have saved the lead, you will find an Escalate button next to the Salesperson field.

Sales teams in OpenERP can be configured in a hierarchical format. Clicking on the Escalate button will transfer the lead up to the parent team from the currently assigned team. This can be very useful in companies that have large sales teams in which junior sales members will escalate leads to more senior sales people as leads are qualified.

**Creating custom stages by the sales team**

For many sales operations, a company may use the same stages for all their sales. However, depending on the complexity of your company's product line, it may be better to create custom stages for different sales teams. For our case study, Silkworm provides creative design services that require a different set of stages while managing the sales process. As we have discussed, the primary purpose of stages in a CRM system is to support a positive customer experience.
To configure the stages for a sales team, click on **Sales Teams** located at the bottom of the **Sales** menu. Click on the appropriate sales team and then on the **Stages** tab to see the current stages assigned to the sales team.

On the far left in the grid, you can click-and-drag the little dot to reorder the stages. The far right of the grid has a small trash icon that can be used to delete a stage.

Each stage is assigned to a related status and you can determine if a stage is applicable to the lead, an opportunity, or to both.

For our new **Creative Designs** team, we have added a few stages to better represent the sales cycle for this service category.
Summary
In this chapter we started by discussing the role of a CRM system in a modern day business. We installed the CRM module, created salespeople, and proceeded to develop a system to manage the sales process. In our example, we walked an opportunity through the various stages in the sales process. Finally, we saw how to modify stages in the sales cycle and turn the opportunity directly into a quotation.

In the next chapter, we will turn our attention to purchasing products and setting up the MRP application to handle production operations.
In this chapter we will start getting into what could be considered the core functionality of most ERP systems. We will begin by setting up a supplier and then purchasing raw material components. After the products arrive, we then receive the products into an inventory and pay the invoice to complete the purchasing cycle.

Topics we will cover in this chapter include:

- Examining a typical purchasing process for a business
- Setting up your suppliers and warehouse locations
- Entering a quote and turning it into a purchase order
- Receiving products from your suppliers
- Paying invoices

Understanding the overall purchasing process
Let us begin by taking a 30,000 foot view of the purchasing process. Putting together a purchasing system requires several steps, and initially it can be confusing for people new to ERP systems. But when you break the steps down and look at them individually the process becomes much easier to understand.

Setting up a supplier
When you set up a supplier you are determining the individuals or companies that are providing you with products. In OpenERP it is perfectly possible to create a product and sell it without implementing a purchasing system. However, to begin using your system for purchasing you will need to configure the suppliers.
The steps you take when setting up a supplier are much the same as setting up a customer. In fact, now is as good a time as any to tell you that OpenERP maintains core customer, employee, and supplier records all in the same model (or table) named `res.partner`. Sometimes suppliers are also referred to as vendors.

### Setting up warehouse locations

Once you have decided to start using OpenERP to purchase your products, then you will need to set up locations to receive the products. In a simple operation you may only have one location. Other companies may have literally hundreds of warehouse locations. In OpenERP each location can maintain its own address, and it is possible to create nested sublocations for better management and reporting of inventory.

### Generating quotations and purchase orders

To acquire the raw product most likely you will be need to create quotations and/or purchase orders to send to your suppliers. In purchasing, these are the documents you create that tell the suppliers which products you require, how many of them you require, and what you expect to pay for those products. Often this process is referred to as procurement. Depending on the industry and the specific location of the company, it is possible there can be a variety of ways to manage quotations and approvals when purchasing products.

### Receiving the product

In a simple purchasing workflow, once your purchase order has been received by the supplier, you will be waiting for them to fulfil the order. At some point you will receive the product. Depending on your industry this could be the same day or as long as several months. When the delivery is complete, you receive the products, and they will move into the location you select. Now that the product has been received, you are ready to create a manufacture order. But first, let's pay the supplier for what we ordered.

### Settling the invoice

Once you have received the product, it is just a matter of time before you must pay for it. Invoicing can happen at the time you order the product, before the product is shipped to you, or after you have received the product. Regardless of when you get an invoice, you can be sure that if you are receiving products, you will eventually be invoiced for them.
When an invoice is received it is essential that it is compared to the purchase order for accuracy. Any discrepancies between the purchase order and invoice must be resolved before an invoice is paid. Essentially this is your mechanism for checking to make sure that you are only paying for the products you have been authorized to purchase. Finally it is good practice to match the receiving or delivery order to the purchase order and invoice as well. This three-way match ensures that you got exactly what you ordered and that invoice reflects exactly what you are required to pay.

**Installing the purchasing application**

OpenERP is a modular set of applications in which you only install the applications you need. Therefore, we must install the purchase management application to continue. By this point you should be familiar with the process of installing a new application to OpenERP.

When you install the purchasing application you will get the following two new menus:

- **Purchases**
- **Warehouse**

The *Purchases* menu is where you can create quotations and purchase orders for the products that you purchase from your suppliers. In the *Warehouse* menu you can manage physical inventories. If you take a few moments to look through the menus, you will notice you can access some of the same features from both menus. For example, you can get to the *Incoming Shipments* view from either menu.
Setting up your first supplier

To begin setting up our first supplier you will navigate to Invoicing | Suppliers and click on Create.

The preceding screenshot displays the Suppliers listing. When there are no supplier records you will get instructions on adding a new supplier.

After clicking on Create, OpenERP will bring up the supplier form for you to fill out.

The Suppliers form
This form is very much like the customer form because it is based on the same basic structure. In fact it is perfectly acceptable for a customer to also be a supplier. Sometimes this can get a little confusing for people new to OpenERP. This chapter will start to make the relationships between companies, contacts, and customers in OpenERP more clear.

**Designating supplier companies versus individuals**

The *Is a Company?* checkbox at the very top of the form is where you inform OpenERP the relationship you have with this supplier. Typically you will be purchasing products from a company. When this box is checked, the *Contacts* tab appears in the section at the bottom of the *Suppliers* form. This allows you to specify contacts for the supplier. If the *Is a Company?* checkbox is not checked then you are specifying that this supplier is an individual.

For our example we will check the *Is a Company?* checkbox.

**Configuring your product for procurement**

When we set up our first product we were only concerned with selling the product to a customer. We essentially named the product and set the price at which we wished to sell it. To purchase the product from our supplier we must provide a little more information. To do this we will edit the product and change the information under the *Procurements* tab.
Navigate to Purchases | Product, then double-click on the White T-Shirt icon to bring up the product form. Then click on Edit to go into the edit mode.

Once you are in the edit mode, click on the Procurements tab.

**Defining the procurement method**

The procurement method is important to discuss in detail. With the standard Purchase Management module installed, you will have the following two procurement methods available:

- Make to Stock
- Make to Order
When you have set your procurement method to **Make to Stock**, it tells OpenERP that when you need this product it will pull it from the inventory. If there is not enough product available, then the delivery order to the customer waits until the stock is available. When you wish to manage your inventory through order points or by making bulk purchases then you will use this procurement method.

When you set your procurement method to **Make to Order**, OpenERP will create a quotation (also known as a draft purchase order) to obtain the required products. If you want sales orders from customers to drive the purchase of this product then use the **Make to Order** method.

In this case, we wish to demonstrate manually creating purchase quotations. Therefore, we want to make sure that our procurement method is **Make to Stock** and our supply method is **Buy**.

**Knowing your supply method**

You have two options for supply method: **Buy** or **Manufacture**. In this example, **Medium White T-Shirt** is our raw product that we will buy. Later when we create the product we intend to manufacture, for that product we will set the procurement method to **Manufacture**. For our raw product, however, we want to **Buy**.

**Setting the cost price of the product**

Most often you will wish to assign a cost to the product. This will be the cost that will appear on your purchase quotations. For our example we have set the cost of the shirt to **9.50**.

**Estimating manufacturing lead time**

This is the average number of days that it takes to manufacture the product. When you have products that are the subproducts of other products, then OpenERP will automatically add together the delays of all the requirements. As we only intend to purchase and not manufacture these t-shirts, we will leave the **Manufacturing Lead Time** at one day.

**Setting records to active**

Like most records in OpenERP, you can set a product to inactive so that they will no longer be available when creating new sales orders or purchase orders.
Once you have any transactions associated with a product, you will not be allowed to delete that product from the system. Instead, set `Active` to false so that the product will no longer appear in the active list.

Assigning suppliers to the product

At the bottom of the `Procurements` tab is the option to add suppliers. It is very common that a company may have multiple suppliers that offer the same product.

Click on **Add an Item** in the **Supplier** grid to add the supplier to the product.

![Create Supplier](image)

Establishing the supplier

You have the choice in the drop-down list to search for suppliers as well as to create and edit a new supplier on-the-fly. To the far right of the drop-down, you can use the small icon to edit the current supplier. In the drop-down list, we have selected **T-Shirt Supply Company** as the company to provide our blank medium white t-shirts.

Designating supplier product name and product code

Because a supplier may use different product codes or product names than your company does to describe a given product, here you have the option to specify how the supplier identifies the product. This information will appear on the purchase quotations and purchase orders you create to make sure you get the right product from the supplier.
Setting minimal quantity
Suppliers will often have a minimum order quantity for a product. Sometimes suppliers may actually sell you a lower quantity but the cost per unit is dramatically higher. Setting a minimal quantity in this form allows you to prevent those problems by forcing purchase quantities to be at least the minimal quantity value. For our example we will set the minimal quantity to 12.00.

Calculating the delivery time
Depending on the supplier a product may take less or more time to obtain. Often this can make a difference when you are putting together a purchase quotation. A product may be cheaper but if the delay is too long and will put the order in jeopardy, you may need to buy the product at a higher price from another supplier who can deliver the product faster. Setting the delivery time in days for the supplier to deliver the product gives your purchasing agents the information they require to make decisions based on price and availability. For our example we have set the delivery time to 4 days.

Creating your first purchase quotation
Now that we have our supplier entered and the product associated with the supplier, we are ready to create our first purchase quotation. This is typically the document you will create when requesting pricing from a supplier (sometimes called a vendor) prior to actually ordering the product. For our example we are going to create a quotation for one dozen **Medium White T-Shirts**.

Navigate to **Purchases | Quotations** and click on **Create** to make a new quotation.
After you click on **Create**, the quotation will appear wherein you can enter the required information.

![Request for Quotation PO00001](image)

When you first create the quotation, the order date will be automatically populated with the current date. You can then select the supplier you wish for the quotation as well as include an optional supplier reference and source document. An example of a source document would be a sales order number that triggered the purchase order.

**Adding products to your quotation**

After you click on **Add an Item**, you can select the product from the drop-down list on the far left. The description will automatically be filled in. The scheduled date will be determined based on the delivery delay from the supplier. We also find the minimum order quantity from the supplier has been pulled to the quotation. Finally, the unit cost is populated from the unit price on the **Procurements** tab of the product.
Printing a quotation and updated status
For now we will skip Incoming Shipments & Invoices and go right to printing our quotation. By default, OpenERP will print to a PDF file. This file can easily be attached to e-mails. Once you have configured an e-mail server you can configure OpenERP to automatically send the purchase order by e-mail.

The status is then updated to show that the quotation request has been sent as shown in the following screenshot:

Promoting status to RFQ (request for quotation) sent
At this point the quotation is considered as being in the hands of the supplier. The supplier could agree to the quotation as it is, or perhaps pricing or other attributes of the purchase order will change. In this stage of the transaction the quotation can be edited as needed. You can add additional items or remove line items that you do not need.

Confirming a purchase order
Once you have a final quotation you are ready to confirm the purchase order. It is very important to understand that once you have confirmed the order, it becomes a purchase order and it can no longer be modified. Once you are sure you wish to finalize the purchase order, click on the Confirm Order button.

If you happen to receive an error message reading No Expense Account when you attempt to confirm the order, check your settings for your chart of accounts. You must have an expense account designated.
After your purchase order has been confirmed, it will be assigned a purchase order number, and the status will be updated.

At this point you are waiting on the supplier to deliver the products and send you an invoice. The status is updated to Purchase Order and is now just one step from the Done condition.

**Receiving products**

If everything goes as planned the products we have ordered will be arriving within four days or less. Once the products have arrived, we must receive our products into the inventory.
Click on Receive Products to bring up the Receiving form.

Getting ready to receive

IN/00003 is sequentially assigned for each transaction. IN is for inventory. When we receive products into the inventory, we create an inventory transaction. You can see the associated purchase order in the top-right corner. We can also see the actual time the order was received compared to the scheduled time of the order. In this case we can see we received the order in plenty of time.

At this point we have not actually received the product yet. This is just showing us the details of the product that we are ready to receive.
Receiving our goods
When you click on the Receive button, you will get a final screen that defaults the quantities from the purchase order.

Once you click on Receive, the status on the product line item is set to Done. The product is now in inventory. Since all of the products on this purchase order have been received, the purchase order status at the top-right corner of the page is set to Received. Finally you will notice the Return Products button that is available for instances in which you need to manage product returns.

If you would like to verify that your goods have been received into inventory, look up the Medium White T-Shirt under Products and click on the Inventory tab. You should notice its quantity has increased by 12,000 units. OpenERP automatically adjusts the stock levels as products are received into your company's inventory. Likewise the stock levels are decremented if products are returned to their supplier.
Paying supplier invoices

Once you have received your product, sooner or later you should receive an invoice. It is of course possible to have an invoice before you receive products. Each business will have to decide the exact workflow for when they pay invoices and under what conditions.

Once you confirm a purchase order, OpenERP will create a draft invoice. You can view this invoice in Supplier Invoices under the Invoicing menu.

The preceding screenshot displays the draft invoice for our t-shirts that we have received. We have the products. Let's go ahead and pay for them.

Click on the line item to bring up the supplier invoice and then validate it.
Now that we have validated the invoice, our remaining options are to pay the invoice or request a refund. For our example we will pay the invoice. Click on **Pay**.

Most of the information will automatically be filled out. The **Payment Ref** field is your tie back to the original purchase order for the product.

When implementing a purchase order system, it is critical to train users thoroughly on how transactions are tied together. While many forms allow you to click on a link to view a related record, fields like **payment ref** store the data just as text. Train and encourage users to quickly use copy and paste rather than re-entering data into search fields.

OpenERP will allow you to configure multiple payment methods. For now, choose **Bank (USD)** and click on **Pay** to complete the transaction.

You have now completed the entire purchasing cycle from purchasing to receiving to finally paying for the product.
Summary

In this chapter we installed the purchasing application and set up a supplier to purchase products. Next, we successfully purchased products and received those products into the inventory. After our products were received into the inventory, we proceeded to pay the invoice to complete the payment cycle.

In the next chapter, we will take the raw materials we have just received into the inventory and use them to manufacture and deliver a finished product. We will create manufacturing orders to define the steps of the production process and allocate the required resources. Coordinating all of your resources, including machinery and manpower, can be a daunting and time-consuming task, but we are learning how OpenERP makes this significantly more manageable.
In this chapter, we will cover how you can use OpenERP to manage the process of manufacturing products. Once you have received the required raw products to your inventory, you can begin manufacturing the end product. Part of the functionality of an OpenERP system is to assist you in scheduling these orders based on available resources. One of the resources is, of course, the raw product. Other resources could include available labor or access to a particular machine. Essentially, the goal is to schedule the manufacturing order at a time that all resources are available and produce the product for an on-time delivery.

In this chapter we will cover the following topics:

- Setting up the manufacturing process
- Defining our bill of materials
- Manufacturing our final product
- Analyzing the inventory report

Creating manufacturing orders

Manufacturing orders define the product you wish to build and the resources that are required to build it. They also designate when you wish to produce the product.
Producing the product
When it is time to actually produce the product, you then inform OpenERP of each of the products produced, and your manufacturing order changes to a status of Complete. In a typical workflow, your raw materials are moved out of the inventory and your finished product is added into your inventory.

Delivering the order
After a product has been produced and has been put into the inventory, it can be packaged and delivered to the customer. Depending on the specific manufacturing environment, a product may not even sit in a physical inventory location at all, and instead may be shipped almost immediately to the customer. Meanwhile in another industry, you may have a product that is produced and then sits in a warehouse for months before delivery. Of course, it is always possible that something gets produced and gets left in dead stock. In this case, you would never have a delivery order and instead use a process to determine how to report that dead inventory.

Defining the workflow for your business
Much like configuring the CRM application, often the most complex part of setting up a purchasing and manufacturing system is not the ERP software itself. Instead, the real challenge is understanding the business requirements and how current processes can best be implemented. If you have never set up a purchasing and manufacturing system before, it is highly recommended that you supplement your knowledge with additional reading on the subject.

A real-world example – producing a custom-printed t-shirt
In OpenERP you manufacture products by creating manufacturing orders. For our example, we will be printing t-shirts that have a custom-designed logo. The basic manufacturing process itself involves using a screen to apply ink to each of the t-shirts. For now, we don't need to know all the details of this process to begin using OpenERP to help schedule and track the manufacturing of the product.

The basic steps in the process are simple:

1. Define a bill of materials that determine what items are needed to produce the final product.
2. Use a manufacturing order to print a design on the blank t-shirts.
3. Deliver the printed t-shirts to a customer.
Installing manufacturing resource planning (MRP)

We must now install the MRP application so that we can begin configuring our t-shirt production. By now, you should begin to understand the modular nature of OpenERP. Install the MRP application just like you did with the other OpenERP applications. Navigate to Settings | Apps.

Creating your first manufacturing order

The flexibility of OpenERP provides a variety of approaches you can take in setting up your system. Manufacturing can also become a complex topic and is one of the more challenging aspects of setting up any ERP system. For our first manufacturing order, we will ignore many of the advanced options.

Keep it simple at first. There are many options and it will take time to understand them all. If you are new to manufacturing systems, it will take you longer to implement OpenERP, and you should consider hiring professional consultants to assist you.

To create your first manufacturing order, go to the Manufacturing menu, choose Manufacturing Orders, and then click on Create.
This is the manufacturing order as it appears just after you click on Create. The MO in the sequential order number that will be assigned stands for, you guessed it, Manufacturing Order. We will use this form to define our manufacturing order to print our custom-designed t-shirts.

Selecting the product
The only product we have entered into OpenERP so far is a blank Medium White T-Shirt. This t-shirt is currently a raw material. Now we want to produce a new product. We must define what our final product will be after the t-shirt has been printed. For our example, it will be Class of 2013 T-Shirt.

OpenERP allows you to create this product on the fly. Just click on the pop up and choose Create and Edit.

The Product and Procurements tabs should look familiar by now. The most important aspect for this product is the supply method. It is set to Manufacture. Only the products that have the supply method of Manufacture can be selected as a product on a manufacture order.
You will also notice that the **Can be Purchased** checkbox is unchecked. This will keep this product from appearing in the product list on a purchase order. We have also put on a **Manufacturing Lead Time** of 3,000 days. This will then impact the dates that are automatically assigned when we create a manufacturing order.

### Building your bill of materials

A **bill of materials** is essentially a list of products that are required to produce another product. You can think of it like the list of ingredients for a recipe. OpenERP needs to know what materials are required for us to produce this Class of 2013 T-Shirt product.

In complex products, a bill of materials can be nested. For example, it may take many products to make a subproduct and then several subproducts to make a final product. For our first bill of materials example, we will be keeping the bill of materials simple. We are just going to require the white t-shirt.

If there are enough white t-shirts, we can print on them. For now, the inks and screens will not be managed in the manufacturing process. This is an example of starting simple and adding more complexity as we build up the system.

Clicking on the **Create and Edit** pop up under **Bill Of Materials** will bring up the **Bill of Material** editor.
Making Goods with Manufacturing Resource Planning

Many of the fields will be automatically filled out, as OpenERP knows we are creating a bill of materials for our Class of 2013 T-Shirt product. In this example, we have added Medium White T-Shirt to the bill of materials. Both quantities are set to 1.000. When we manufacture one Class of 2013 T-Shirt, we will require one Medium White T-Shirt. Often, a bill of materials will contain multiple items.

Confirming production

Once you click on Confirm Production, you are ready to manufacture the product. OpenERP will provide reasonable defaults, which you can override as required. When production is confirmed, the production of the product has not happened yet. It has only been confirmed for production and all is ready to go. You can tell we are ready to go because the Produce button is available.

Here is what our Class of 2013 T-Shirt manufacturing order looks like now:
Producing the product

After you click on the Produce button, you will be prompted to confirm that the product has been produced.

The Mode option allows you to choose if you want to consume the raw materials and produce the final product, or simply consume the raw materials. The latter option may be used in a long production cycle where you need to show the raw materials gone from the inventory but you want another step to actually produce the final product. For this example, we will choose Consume & Produce to consume and produce the final product. Click on Confirm.

The product has now been produced and is available in the inventory for sale. Congratulations, you have just used OpenERP to manufacture your first product!

Analyzing the inventory

One of the easiest ways to see the effect of our manufacturing order is to look at the inventory analysis. Navigate to Reporting | Warehouse | Inventory Analysis.
Here you will see that we now have one **Class of 2013 T-Shirt** and eleven units of the **Medium White T-Shirt**. The inventory has accurately reflected the purchases we have made as well as the products consumed and produced by our manufacturing order.

### Managing routings and work orders

This first manufacturing order was very simple, and our bill of materials only contained one product. In many companies the manufacturing operations are far more complex. For example, in some instances depending on the attributes of the product, the manufacturing process could involve different work centers or alternative steps to produce the final product. By default, OpenERP’s manufacturing application takes a more simplified approach. Going into the settings of the manufacturing application allows you to specify additional options.
Simply go to the Settings menu and select Manufacturing under the configuration section on the left. Here under Manufacturing Order | Planning, you can check Manage routings and work orders. Once this option is checked, you will have the ability to manage more complex manufacturing processes inside OpenERP.

After you apply the changes by clicking on Apply, the menus will refresh and new options will be added to the manufacturing application.

Creating a work center
In our previous simplified manufacturing order, we simply specified the raw product required in a bill of materials and then turned that into a finished product. Now we will expand this example to specify the human labor that goes into printing our Class of 2013 T-Shirt. In OpenERP we define the parameters in a work center.
For the purpose of our example, we will create a work center named Printing that is responsible for taking the blank t-shirt and applying the design to create the final product. We begin by going to the manufacturing application, and under the Configuration menu choosing the Work Center option. Then, we click on Create to set up a new work center record.

In our example, we have named the work center Printing. In a full implementation, it would be common to have different work centers based on the work performed.

**Defining the resource type**
While setting up a work center, you are required to specify Resource Type. This setting can either be Human or Material. As you may expect, a human resource will primarily depend on human interaction in performing the work, while a material resource would typically indicate a nearly automatic machine that once configured will perform the work unattended.

**Setting capacity information**
While defining a work center, it is possible to define Capacity Information that will allow you to estimate the cost and time required to produce your products. In our example, we are going to configure this work center so that we can estimate the time required to produce a t-shirt. Let’s look at each of the capacity values.
Efficiency factor

The Efficiency Factor field is a metric of how efficient this work center is at completing tasks. Often, the efficiency factor is most valuable in allowing you to tweak your work center capacity without modifying many of the other variables. If, for example, you have an efficiency factor of 2.00 (or 200 percent), the work center will complete twice as many tasks. For our example, we are leaving the efficiency factor as the default of 1.00 or 100 percent. Some consider it lazy work center design to modify efficiency values rather than more accurately configuring other capacity settings.

Capacity per cycle

The Capacity per Cycle field allows you to determine how many tasks the work center can do in parallel. For example, if you had a work center that could be configured with three workers and all three workers can complete a cycle at the same time, you could set Capacity per Cycle to 3.00. When a manufacturing order is then routed to the work center, the work center can complete three tasks at the same time. For our example, we will assume one worker and therefore, one capacity per cycle.

The time for one cycle in hours

Time for 1 cycle (hour) specifies how much time in hours it takes to complete one cycle. In our example, we are producing t-shirts. Therefore, this value indicates how long it takes to produce one t-shirt. In this example, we have specified that each t-shirt will take two minutes (00:02) to be produced by this work center.

The time before and after production

Many work center operations will have time required for setup and tear-down times before you can begin actually producing the product. This is certainly true for our example. It takes time for someone to prepare a printing press before the first t-shirt can be printed. For our example, we have estimated five minutes of setup time. Likewise, when we are done producing the last product in our work order, it takes time to clean up and prepare for the next job. In this example, we have estimated five minutes (00:05) of time at the end of production for clean-up operations.

Costing information

If desired, you can also specify a product that is produced by a work center. In some manufacturing operations, a work center will always produce the exact same assembly or subassembly. For our example, we want our work center to be flexible, and it may print any number of finished products; so we will leave the work center product field blank.
Creating routing orders

After defining a work center, we need to define a way to specify under which conditions we should use the work center. This is accomplished by defining routings. For our example, we are going to keep it simple and use routing to send our manufacture order to the printing work center for the finished product to be produced. In a real-world example, the job may use routings to go through many work center operations before the final product is produced.

To create a routing order, go to the manufacturing application and choose **Routings** under the **Product** submenu. Click on **Create** to bring up the new routing form.

![Create Routing Form](image)

In our example, we have named the routing **Print Job** and specified **Production Location**. For complex routings, you can specify the sequence of the operations. We could, for example, have a **Design** operation and a **Build Screen** operation before the **Print Job** operation. Then, we could specify a **Quality Assurance** operation and a **Packing** operation after **Print Job**.
Creating a manufacturing order with routing and work center

Now that we have defined our work center and our routing operation, we can create a manufacturing order that will utilize our new production steps. In this example, we are going to produce 15 Class of 2013 T-Shirts.
You will notice that in the manufacturing order, we have selected **Print Job** for the routing of this order. This is the key field that will send this job to the printing work center to be produced. Clicking on **Confirm Production** will load in the products that will need to be consumed, as well as the work orders required to produce the product.

Notice on the far right that it is estimated it will take 0.67 hours to produce these 15 Class of 2013 T-Shirts. How did we arrive at that number? It is a result of the capacity information we specified in the **Printing** work center earlier in the chapter. The analytical formula for this would be:

\[
\text{Capacity information} = 5 \text{ minutes setup} + (2 \text{ minutes} \times 15) + 5 \text{ minutes tear-down time}
\]

**Producing the manufacturing order**

Now you can click on **Produce** to send the manufacturing order to the work center and produce the 15 t-shirts. You will get a confirmation screen to confirm that you wish to create all 15. After production, the state will go to **Done**, and you will find the 15 t-shirts in your inventory ready for shipping.
Summary

In this chapter we installed the MRP application to begin setting up our manufacturing process. A bill of materials was created to define what products would be consumed when our product was manufactured. Finally, we manufactured our final product and looked at the inventory analysis report to verify our results.

In the next chapter we will take a closer look at accounting and other reporting options. Setting up your chart of accounts is an important step that we'll cover, as well as reviewing journal entries, creating invoices, and receiving payments. We will also be defining sales taxes and managing fiscal periods. Yes, there is a lot more to cover!
Configuring Accounting and Finance

One of the nice things about OpenERP is you can get up and running fairly quickly without having to spend a lot of time setting up complicated accounting and finance options. OpenERP does a pretty good job of creating basic charts of account structures as a point to get started and for getting familiar with OpenERP. While setting up a production system for your company, however, you will want to take time to properly define your accounting requirements.

In this chapter, we will learn the following:

- Setting up the chart of accounts
- Reviewing journal entries
- Creating customer invoices
- Receiving payments
- Using OpenERP's reporting options

Defining the chart of accounts for your business

The backbone of an accounting system setup is the chart of accounts. Wikipedia defines a chart of accounts as follows:

A chart of accounts (COA) is a created list of the accounts used by a business entity to define each class of items for which money or the equivalent is spent or received. It is used to organize the finances of the entity and to segregate expenditures, revenue, assets and liabilities in order to give interested parties a better understanding of the financial health of the entity.
It is very likely that if you are setting up OpenERP for an existing business, you will be asked to configure the chart of accounts in OpenERP to match the account structure the business is already using. Even if you are not tied to any existing chart of accounts, it is inevitable that you will need to have a firm understanding of how the accounting functionality in OpenERP works if you are going to have a successful implementation.

If you are completely unfamiliar with accounting, then this chapter may prove somewhat challenging. It is important to get familiar with accounting basics if you are going to be successful in implementing any ERP system. Here are a few resources to help you get started.

### Installing the Accounting and Finance application

OpenERP configures a basic accounting structure when you install base modules, such as Sales and Purchasing. To access all of the accounting configuration options, you must install the Accounting and Finance module. If this module is not already installed in your configuration, navigate to Settings and click on Apps to pull up the available applications. Look for Accounting and Finance and click on the Install button to install the module.

![Accounting and Finance](image)

After you have installed the Accounting and Finance application, your menu structure at the top of OpenERP will change. Before installing the module, you likely had an Invoicing menu that contained the necessary options for the Sales and Purchasing applications. Once the Accounting and Finance module is installed, the Invoicing menu is replaced with an Accounting menu and is populated with several more options.
Viewing the current chart of accounts

We will begin by learning how to view the current chart of accounts in OpenERP. Go to the Accounting menu and choose Chart of Accounts under the Charts submenu. You will be presented with a screen that lets you select Fiscal year and Periods for which you wish to review the Chart of Accounts.

OpenERP will default Fiscal year to the current year, and Periods will default to periods through the current month in the fiscal year.

Click on Open Charts to view the chart of accounts for the specified time frame.

In the previous screenshot, we see the currently configured Chart of Accounts in a hierarchical tree structure. The Liabilities accounts have been opened to demonstrate how OpenERP nests accounts inside each other:

2 - Liabilities

20 - Current Liabilities

2000 - Payable

200010 - Account Payable
The ability to nest accounts inside one another allows reports to roll up totals so you can analyze the financial status of the company at any level. For example, you can look at the total of all Liabilities or only look at the liabilities you currently have in Account Payable.

How were transactions created in Account Payable?
Looking at the Chart of Accounts, we can see that we have transactions in the Account Payable liability account. Account Payable typically includes the total of current invoices you have open from your suppliers. When you purchase goods and you receive an invoice, Account Payable will increase by the amount of the invoice. When you pay the invoice, the Account Payable account will decrease by the amount you pay on the invoice.

To see the transactions that created the $114.00 in Account Payable, double-click on the 200010 account.

When we open the account, we see all the transactions that were involved. In the first column we see the Journal for the transaction. The Purchase Journal (USD) entry specifies PO00001 as both the Name and Reference for the transaction. On the far right of the entry, you will see the Credit amount of $114.00. This was the amount of the purchase order, and Account Payable was credited $114.00 to account for the liability that the company owed from purchasing the product.

When the invoice for PO00001 was paid in full, OpenERP created the journal entry to debit the Account Payable account to reflect the $114.00 payment.
Viewing the other journal items

In the top-right corner the OpenERP search box, we can see that the view is limited to the Account Payable account. If we click on the small close box on the Account Payable filter in the search view, we can see all the journal items.

If you look carefully at the new list of journal items, you will see there are two additional transactions.

One of these transactions is a $114.00 debit to the 529000 Purchases – Resale Items account. As we discussed previously, when the purchase order was invoiced, OpenERP credited $114.00 to Account Payable. At the same time, OpenERP creates the $114.00 debit to the 529000 Purchases account. This is known as double-entry accounting. The result is that we can look at account 529000 to see the total of our purchases.

When we paid the invoice, OpenERP created the $114.00 debit to Accounts Payable to account for the fact the company no longer has $114.00 in accounts payable liability. At the same time, OpenERP credited the account 100002 Bank for $114.00 to follow the money leaving that account.

If this is still a little confusing, don't worry. We are now going to follow a set of transactions from the accounts receivable side so you can better understand how OpenERP handles accounting transactions.

Tracking accounting transactions through the sales and accounts receivable process

In the previous example, we were looking at the Chart of Accounts and determining what transactions created the entries. Next, we will sell an item to a customer and see exactly how that transaction affects the accounting entries in the journal.
Let's begin by creating a new sales order.

Go to Sales and click on Sales Order to bring up the sales order listing. Click on Create to create a new sales order as shown in the following screenshot:

If you have followed along with our examples, then you will already have the customer and product entered to create the sample sales order. Otherwise, you will need to add a customer and a product if you wish to follow along on your computer. In this example, we have created a sales order for five medium white t-shirts.

At this point, if you were to go and look at the Chart of Accounts, you will not see any changes to the transactions. Why is this? The way OpenERP is currently configured, we must manually create an invoice. Only when we click on the Create Invoice button at the top of the screen will OpenERP actually create accounting transactions.
Click on **Create Invoice** to generate a draft invoice for this sales order as shown in the following screenshot:

Next, choose to invoice the whole sales order and then click on **Create and View Invoice** as shown in the following screenshot:

Because this is just a draft invoice, if you look at the **Chart of Accounts**, you will see no changes. However, if you look at the **Draft Invoice**, you can see exactly the transactions that will be created once you validate the invoice. Notice in the header of the document that the account field reads **120010 Account Receivable**; this account will be debited the amount the customer owes to the company once the invoice is generated.

In the line item of the purchase order, you will see the **479100 Sales** account. This will be the account that will be credited for the sale of the medium white t-shirts.

Click on **Validate** to post the invoice and create transactions.
Viewing the transactions created by validating the invoice

Now that we have validated our invoice, OpenERP has automatically created the accounting transaction to increase our **Accounts Receivable** assets and the accounting transaction to record the sale.

![Chart of Accounts](chart_of_accounts.png)

If you take a close look at the **Chart of Accounts** after posting the invoice, you will see that the **120010 Account Receivable** account has been debited by $90.34 to show the new current asset representing the customer invoice. The customer owes the company $90.34. As you create invoices, more customers owe you money, and as a result, **Accounts Receivable** will continue to grow.

Next, notice the **479100 Sales** account has been credited with $90.34. This account will continue to be credited for products you sell.

For our example, we are using only one sales account to keep things simple. In most companies, you will have far more sales accounts to organize the various types of products sold.

Now let us see what happens to these accounts when a customer pays their invoice.
Navigate to **Accounting** and select **Customer Invoices**, then click on the invoice to bring up the form. Click on **Receive Payment** to bring up the **Pay Invoice** screen as shown in the following screenshot:

OpenERP will automatically fill in the **Customer**, **Paid Amount**, **Date**, and **Period** fields. For our example, we have selected the **Cash (USD)** payment method. You have the option to provide both a **Payment Ref** and a **Memo** to document that invoice payment.

Click on the **Pay** button to pay the invoice and create the appropriate accounting transactions.

The invoice is now paid and the journal entries have been automatically created. However, when it comes to money going in and out of the company, extra care is taken to make sure there is an opportunity to reconcile cash and bank transactions. For example, the **Accounts Receivable** clerk may receive a check from a customer and mark the invoice as **Paid**. But how do we know for sure that the customer’s check actually made it into the bank account?

For this reason, the journal item for the cash received to pay the invoice is created in an **Unposted** status. Let us view the journal entry.
Configuring Accounting and Finance

Under the **Accounting** menu, choose **Journal Entries** to bring up the list of entries in our general journal as shown in the following screenshot:

In the list you will notice we have two **Unposted** entries. One is the entry from the payment of the purchasing invoice of $114.00 that is to be taken from the bank account. The other entry is the cash customer payment of $90.34. Let’s go ahead and post the cash entry for Mike Smith’s invoice.

Click on the unposted cash payment for $90.34 to pull up the journal entry.

In this cash receipt, you will notice that we can see the details on exactly which accounts will be affected when we post the entry.

- 120010 Account Receivable is credited with $90.34. This will reduce this asset account.
- 100001 Cash is debited with $90.34. This will increase this asset account.
Essentially, this journal entry transfers the potential asset the customer owes the company into the cash account. The customer's account is adjusted to reflect their payment.

Click on Post to post the entry.

Now that we have posted the entry, let's take another look at the Chart of Accounts to see how the Cash and Account Receivable accounts have been changed to reflect the transactions as shown in the following screenshot:

First, notice that now our Account Receivable (120010) balance is zero. We can see the debit of $90.34 that was created when we invoiced the customer. Then, we can see the $90.34 credit that was posted when we received the customer payment. Now, take a look at the Cash account (100001). It has a debit and a balance of $90.34. This is the cash that we have received from the customer.

**Practice posting transactions and tracking the results**

Remember that people spend many years and even get full degrees in financial accounting. It is important that you take time learning how each process you implement affects the accounts in OpenERP. When implementing an OpenERP system for your company, take the time to get this right. It will save you a lot of pain in the long run.

**Where are my taxes?**

If you have been looking at the journal entries and following along, you may have noticed that the sales taxes are not reflected in the Chart of Accounts. By default, OpenERP posts the entire transaction to the Sales account. This was likely done in order to make OpenERP a little easier to implement for people that are new to accounting. You don't have to trouble yourself with setting up special tax accounts. Fortunately, OpenERP provides a very easy way to see your sales tax liability.
Configuring Accounting and Finance

The Chart of Taxes appears right under the menu item for Chart of Accounts as shown in the following screenshot:

<table>
<thead>
<tr>
<th>Tax Case Name</th>
<th>Case Code</th>
<th>Period Sum</th>
<th>Year Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tax Balance to Pay</td>
<td></td>
<td>-7.84</td>
<td>-7.84</td>
</tr>
<tr>
<td>Tax Received</td>
<td></td>
<td>7.84</td>
<td>7.84</td>
</tr>
<tr>
<td>Tax Received at default rate</td>
<td></td>
<td>7.84</td>
<td>7.84</td>
</tr>
<tr>
<td>Tax Received Rate X (Exempt)</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Tax Paid</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Tax Bases</td>
<td></td>
<td>82.50</td>
<td>82.50</td>
</tr>
</tbody>
</table>

In this tax report, you can see the Tax Received totaling $7.84 from Mike Smith's t-shirt order. This set up will work for some companies, but many companies are going to want to have their sales tax payable represented in their chart of accounts.

### Specifying the account for your sales tax

As we have discussed, by default, OpenERP does not configure sales tax to post to a specific general ledger account. To configure OpenERP to create the appropriate journal entry in Sales Tax Payable, open up the tax entry form.

Under the Accounting menu, choose Taxes under the Configuration section at the bottom of the menu. Then click on the tax that you would like to configure. In our example, we have opened up the 9.50 percent sales tax rate.

As you can see, looking at the form in the preceding screenshot, there are a lot of configuration options for taxes. To configure OpenERP to post to a specific account, we are concerned with the Invoice Tax Account value located directly under Invoices. Here, we have specified to send our taxes to the 255000 Sales Tax Payable account when an invoice is posted. With this configuration, OpenERP will now credit the taxes to this account.
As an example, we created another invoice for Mike Smith that resulted in $3.14 in taxes. We can now see this tax represented in the chart of accounts in the **Sales Tax Payable** account as shown in the following screenshot:

### Setting up your own accounts

To this point, we have used the standard United States Chart of Accounts template provided by OpenERP. Most companies, however, will need to modify this chart of accounts or even set up an entirely different chart of accounts to match the needs of their business. As an example, we are going to add an additional sales account specifically for t-shirts so that we can better organize our sales into types of products.

To set up a new account, navigate to the **Accounting** menu and then down to the **Configuration** section and select **Accounts**. OpenERP will present you with a listing of all your current accounts in OpenERP. Click on **Create** to add a new account as shown in the following screenshot:
Notice in the preceding screenshot that we have specified the account code as 479110. Why did we choose this as the account code? OpenERP had already provided 479100 for the general Sales account. Therefore, 479110 was an appropriate account code to choose for our t-shirt sales. For the name of the account, we named it simply T-shirt Sales.

In addition to specifying the Account Code and Name, we must set the Parent account. Even though this is not a required field, you need to be careful and select the appropriate parent for the account. In this case the parent account is 40 Income. This will group the t-shirt sales in with the income of other products the company sells.

The other important setting is Account Type. OpenERP needs to know the type of account you are setting up. For example, if you were setting up an account that was to track costs of products you must purchase to produce your products, you would specify an Expense account type.

Take time planning your chart of accounts in OpenERP. Even if your company has already been using an existing chart of accounts, it is always a good idea to evaluate the current chart of accounts and make any improvements given the current state of the business.

Specifying a new account for your product category

With OpenERP, you can manage accounts at the product category level. Therefore, all products under a given category can utilize the same account settings. Let's create a new product category, T-shirts, for our medium white t-shirts, and assign that category to the 479100 T-Shirt Sales account we created. Later, we can add all t-shirt products under this category.
Navigate to the Sales menu, and in the Configuration section, select Product Categories under the Products submenu. This lists the current product categories as shown in the following screenshot:

At this point, you will see we only have two categories. Click on Create to add a new category for our t-shirt products.

We have named our new category T-shirts. All t-shirt products can now be grouped under this category. Also notice that we have a Parent category set for All products / Saleable. This allows you to view the t-shirt products along with all the other products when you choose the Parent category.
Most importantly from an accounting standpoint is we have assigned 479110 T-shirt Sales to Income Account, which we have set up in the chart of accounts. When an invoice is posted that has a line item attributed to this product category, the amount for that line item will be posted to 479110 T-Shirt Sales. For Expense Account, we have specified 529000 Purchases – Resale Items that was provided in the built-in template.

An exercise for you! Try going into the product record for the medium white t-shirt and set the Product Category to T-shirts. Create a sales order, turn it into an invoice, and validate it. View the chart of accounts and you will see the income for your t-shirt in the specified income account.

Remember that it is important to practice using OpenERP until you are comfortable setting up accounts and understand clearly where the transactions are posted. A little bit of time and effort put in during the configuration will save you a lot of time later.

**Configuring your fiscal year and periods**

Companies are typically required to report financial information on a monthly, quarterly, and yearly basis. In configuring OpenERP, you must specify the fiscal periods that your company uses for reporting. By default, OpenERP configures your installation with an accounting period for each month of the year. OpenERP also provides an open period that is used for common beginning of the year adjustments. While many companies have a December 31st closing date for their year, there are quite a few companies that have fiscal years that begin and end on other dates. There are many reasons why a company may have an alternative fiscal year structure. In OpenERP, we can view and edit the accounting periods by going into the Configuration section of the Accounting menu and selecting Periods.

As you can see, each period is defined with starting and closing dates. In this view, you can modify the periods to meet the needs of your business. On the far right of the listing, you will notice a Status column. This tells you if the period is open or closed.
Closing a period

When you have completed all the transactions for a given period, it is a good idea to close that period to prevent any additional postings. This is important because you do not want future transactions you enter to accidentally post to a previous period because of user error. To close a period, simply click on the period you wish to close and click on the Close Period button in the form. You will be presented with a form to confirm that you wish to close the period.

Once you have closed a period, it is possible to reopen the period if you must post a transaction to that period. Simply click on the period again and click on the Re-open Period button.

Creating journal entries

While OpenERP will create many journal entries automatically, when you perform various operations in the system, it is inevitable at some point that you (or your accountant) will wish to create a manual journal entry. A manual journal entry allows you to adjust account balances in a way that can easily be tracked and audited.

For our example, we are going to create a journal entry that will account for a small investment by one of the company owners. When someone puts money as an investment into a company, they are not buying anything and they are not selling anything. While there are other potential methods, a simple journal entry is a straightforward way to accurately record the transaction.

To enter a journal entry, navigate to the Accounting menu and select Journal Entries in the Journal Entries section and click on the Create button.
The **Journal Items** screenshot is displayed in a saved mode because OpenERP does not adequately display all the information when you are editing. First when you create a new journal entry, you will need to pick a journal for posting. In our situation, we could have posted to an opening balances journal or to the bank journal. We are considering the $5000 investment after we have already been doing business for a while, so we posted to the bank journal. Next, you can select the period for the journal entry.

Whenever you create a journal entry, you will add at least two line items. Furthermore, the line items must balance out one another. In our example, we are basically putting $5000 from the owner into the company bank account. This is specified in the first line item as **100002 Bank** and the $5000 debit.

Typically, any investment the owner puts into the company must also be recognized as a liability for the company. Why? The money really does not belong to the company. Instead, the $5000 in this case is considered the **owner's equity**. The owner is entitled to get that money back, and therefore it is booked as a liability. You can verify this by opening up the chart of accounts and looking at the list of main accounts. **Liabilities and Equity** are grouped together and then divided out as you drill down into the account hierarchy.

OpenERP sets up a **Capital Stock** account that allows us to post the $5000 we have put into the bank as capital stock for the owner. Once you save your journal entry, it is in a draft form. To post the journal entry and have it appear in your chart of accounts, you must click on the **Post** button.

After we have posted, we can look at our balances in the chart of accounts. You will notice that our assets now include the $5000 investment, and the Capital Stock in the **Liabilities and Equity** section of the chart of accounts represent the liability the company now has to the owner who did the investing as shown in the following screenshot:
Summary
In this chapter, we examined how OpenERP generates transactions and how you can use the chart of accounts to look at how those transactions originated. We examined both the **Accounts Payable** and **Accounts Receivable** accounts and how an invoice is posted.

In the next chapter we will look at the Human Resources application. HR allows us to keep track of employees, the hours they work, and the services they provide. Staff can be assigned to user-defined departments and designated as managers of other employees. And since your employees will often times also be users of the OpenERP system, the HR module is tied tightly to the user administration system, which manages access rights and messaging.
Implementing the Human Resources Application

Over the past few decades, companies have had increasing demands placed on them to keep track of employee-related information. OpenERP has a variety of modules that can help your company organize information involving your employees. Some of these applications that track things such as time and attendance can become a critical process that helps a company contain costs. In this chapter we will look at how you can integrate human resources (HR) applications.

In this chapter we will look at the following topics:

- Maintaining an employee directory
- Submitting and approving timesheets
- Granting leaves of absence, time off, and holidays
- Recruiting and hiring personnel

Taking a modular approach to human resources

Like the rest of OpenERP, the human resource applications allow you to implement the functionality you need today, and then later add additional modules. This approach makes it much easier to start using OpenERP right away to address specific company needs. The best way to be successful with implementing systems is to plan ahead and implement in stages. Once you are successful in putting one application in place, you can then move on to putting additional applications in place.
Implementing the Human Resources Application

Installing the Employee Directory application

When you install the base OpenERP applications, you get the ability to manage system users, but you will notice that there are no menu options for entering and managing employees. To begin working with human resource applications, you will need to install the base Employee Directory application.

Go to the Settings menu and install the Employee Directory application using the same process as the previous OpenERP applications. Click on the Install button.

![Employee Directory](image)

After you have installed the Employee Directory application, you will see a new menu at the top: Human Resources. Clicking on the menu will take you to the employee listing where you can begin adding employees.

If you have added users such as the administrator account, they will be considered employees and included in the list. All users are employees, but not all employees are users.

Creating a new employee

Clicking on the Create button will bring up the form for you to begin entering a new employee into OpenERP.
The only required field in the employee form is **Name**. All the other fields are optional. OpenERP will default the working address to be the same as the company address. While most fields are self-explanatory, we will go over several of the more important fields to take into consideration.

**Employees' public information**

Let us first have a look at the fields under the **Public Information** tab.
Implementing the Human Resources Application

Related User
In the Contact Information section, the Related User field will allow you to associate the employee with a user account in OpenERP. Simply click on the drop-down list and choose the user you want associated with the employee. It is also possible to add users on-the-fly by choosing Create... from the Related User drop down.

Visibility
The Visibility field can be set to Public or Private. OpenERP allows you to set up a portal for your company to maintain and allow access to the employee directory. The Visibility selection allows you to determine if this employee's information should appear in the portal.

Department
The employee department is a common way a company will organize employees. In our example, we are going to create a production department.

In this screen we have named the department Production and set Manager of this department to Tina Robbins. Also, you will notice the Parent Department field. This field allows you to create a hierarchical structure of departments for your company. Typically, you will wish to look at the organizational chart of a company and take some time in preparing the company department structure.
Job
The field named simply Job allows you to manage job titles for employees inside OpenERP.

![Create Job Form](image)

Here in this screenshot, we have created a job title for Tina Robbins, the production manager. As you can see from this form, job titles are tied to departments. This means that to properly configure OpenERP, you would need to create job titles across departments. This means that you do not necessarily want to name a job simply as manager; while looking at the list of job titles, it would make it difficult to know which department that manager may be associated with.

Manager and Coach
The Manager and Coach fields in the employee screen can be used to specify any other employees that are already in OpenERP. The manager is often called the supervisor in some companies. The Coach field is just an optional field that you could use to specify another relationship the employee has that is valuable to the position.
Employees' personal information

The second tab on the employee screen, Personal Information, contains the personal information for the employee.

The Nationality field allows you to select from the entire country listing that comes preloaded in OpenERP. Typically, the Identification No field would be used for an employee badge. OpenERP includes the Passport No field on the form, which may be required in some cases where a company is required to report citizenship information to the government. The Other Id field can be used to collect any other information that may be part of the human resource record for the employee.

If you do decide to enter a home address in the Home Address field for the employee, you will be taken to another screen. Near the bottom of the form you have the ability to specify Gender, Marital Status, and Date of Birth for the employee.

Under HR Settings, the only field on the form after installing the Employee Directory applications is a field to determine if the employee is active. If you wish to make an employee inactive, simply uncheck the Active checkbox.
Managing timesheets
OpenERP allows you to install a human resource module that will allow you to track employee time and attendance. Timesheets are most useful when you have jobs for which you need to account hours and assign those hours to projects or customers. To utilize this feature, install the Timesheets application.

Once you have installed the Timesheets application, the Human Resources menu will be expanded to include a new section called Time Tracking. This is in the section that you manage and validate timesheets.

Looking at your current timesheet
We can begin by clicking on My Current Timesheet to bring up your current timesheet. If you are logged in as Administrator, you can choose to view any timesheet. In this case, we are viewing the timesheet for Tina Robbins and have clicked on the Add a Line button to add a line to the timesheet.

In OpenERP, you will need to set up an account to track time and attendance before you can begin entering hours into your timesheet.
Implementing the Human Resources Application

Setting up an account for tracking time

When you begin tracking time, it is important for you to consider how you want to report that time and if you would like to tie the time employees spend to a specific customer. For our example, we need to set up an account to track our art production time. When you click on Add a Line, you will be prompted to select an account. To add an account, select Create from the small pop-up menu after you have added a line to the timesheet.

An analytical account allows you to track costs and other financial information separately or from a different perspective than your standard accounts. For the most part, managers and directors are going to be interested in analytical accounts as opposed to the financial department that will be focused primarily on general accounts.

After a little bit of configuration in setting up the account, OpenERP can take care of the rest once we enter the hours into the timesheets.

In the previous screenshot, we have set Account/Contract Name to Art Production and have set Type of Account to Analytic Account. There are four options you can choose for the types of accounts to use:

- Analytic View
- Analytic Account
- Contract or Project
- Template of Contract
We choose to use **Analytic Account** because at this point in our example, we simply want to track the number of art production hours put in by employees in the department. If you wish to track work by contracts or projects, you can either select **Contract or Project** for predefined contracts and projects or select **Template of Contract** for a reusable template containing default data. Choosing these options will display additional parameters to associate with the time allocated to the account.

![Contract Information](image)

OpenERP has the capability to specify a date range for contracts and automatically notify the account manager when it is time to contact the customer to renew. **Timesheet Invoicing Ratio** lets you specify how much of the invoice is paid up front and how much is paid out during the remaining period of the contract. Use the text area in **Terms and Conditions** to specify any details on the contract requirements.

**Configuring the employee for entering timesheets**

There is one more step we need to take before we can begin entering timesheets for an employee. When we installed the **Timesheets** application, additional options were added to the **HR Settings** section of the **Employees** form.
Implementing the Human Resources Application

An entire new section named Timesheets is added with the ability to select a product from the Product field to associate with the employee's time as well as an analytic journal from the Analytic Journal field that organizes the timesheets.

If you do not specify the analytic journal for the employee, you will not be able to create a timesheet and submit it to a manager for approval.

Tracking employee hours with the timesheet

After you have set up the account, you can enter hours into the timesheet. For more complex timesheet requirements, you can create additional accounts and then add as many lines as you need to properly account for all the hours worked.

Here is the timesheet after the week has been filled out for Tina Robbins:
You can enter additional details on the work performed by clicking on the **Details** tab.

![Image](image1.png)

In addition to specifying a description or changing the analytic account the time needs to be attributed to, you can also determine what percent of the time on the timesheet can be invoiced. This will allow you to manage invoices in which the labor time of the employee is part of a contract.

After the hours are entered, the timesheet can be submitted for managerial approval by clicking on the **Submit to Manager** button in the top-left corner.

**Validating timesheets**

Once a timesheet has been submitted to a manager, the timesheet will appear under the **Timesheets to Validate** section for that manager when they log in to OpenERP.

![Image](image2.png)

To validate a timesheet, simply click on the timesheet you wish to validate and click on the **Approve** button. If you wish to decline a timesheet, click on the **Refuse** button instead.
Leave management

In addition to managing and approving daily timesheets, it is also possible to install an OpenERP human resources application that will manage holidays, leaves, and other information related to employee time off. We install Leave Management the same way as the other OpenERP applications by clicking on the Install button.

After you have installed the Leave Management application, you will have a new section added under the Human Resources menu: the Leaves section. The primary purpose of this application is to provide an easy mechanism for employees to request leave and for their managers to approve or deny the request.

Creating a leave request

When you click on the Leave Requests menu option, you are taken to a calendar that will show you your current leave requests. Naturally, if there were no prior leave requests made or there are none for the current month, then the calendar is empty.

Click on a day in the calendar to tell OpenERP to schedule a leave request beginning on that day.

In this example we have requested a leave of absence of three days.
Leave Type
For our example we have chosen Leave Type of Compensatory Days. This implies that the employee is taking this leave with pay. Alternative leave types, such as Legal, Sick, and Unpaid, can be managed for reporting purposes.

Duration
When we change Duration using the date fields, OpenERP automatically recalculates the days field for us.

Mode
The Mode field deserves special explanation and dramatically changes the way in which this leave request is submitted. For our example, we are submitting the leave request for a single employee. If the By Employee mode was chosen instead, we could submit leave requests that match all the employees who share the same employee tag. This can be useful if you need to schedule leave for entire sets of employees.

Employee
This field lets you set the employee for which the leave request is made.

Department
This field lets you set the department relevant to the leave request.

Submitting for approval
When requesting leave, clicking on the Save button is all that is required to save the information and send it on to the assigned manager for the employee for approval.
Implementing the Human Resources Application

Approving leave requests
Clicking on the Leave Requests to Approve option in the Leaves section of the Human Resources menu pulls up the list of leave requests to approve. In our example, we see the leave request we have submitted for Tina Robbins.

Much like the timesheet approval, you simply click on a request and then click on Approve to accept the request or Refuse to deny the request.

The Leaves Summary page
To see all the leaves that have been approved, click on the Leaves Summary option under the Leaves section.

By default, this report groups by Leave Type. Using the grouping and filtering options of OpenERP, you can configure Leaves Summary to display just the information on leaves you require.
Managing the recruitment process

Many human resource departments can spend a great deal of time managing the recruitment process. OpenERP provides an application that can help organize the information and make it easier to keep track of the communication required while hiring new employees.

Install the Recruitment Process application as you have done for other OpenERP applications.

After the Recruitment Process application has been installed, OpenERP will add an additional section to the Human Resources section as well as new options under the configuration menus at the bottom.

Defining recruitment stages

The OpenERP Recruitment Process application organizes the recruiting process around stages. This is much in the same way that an opportunity in the CRM application is organized around stages. The goal, of course, is to find new employee leads and then convert them into company employees. To look at the stages OpenERP sets up by default, go down to the Configuration section under Human Resources and choose Stages.

Stages can be created, edited, and deleted just like other records in OpenERP. OpenERP also allows you to specify that some stages are specific to a given department. Let us implement this with a practical example. In our company we are going to say that the second interview is only required for the sales department.
Implementing the Human Resources Application

You can perform the following small exercise:

1. Add a new department named Sales.
2. Restrict the Second Interview stage to this department.

Here are the resulting stages after the previous changes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Specific to a Department</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Qualification</td>
<td></td>
<td>New</td>
</tr>
<tr>
<td>First Interview</td>
<td></td>
<td>In Progress</td>
</tr>
<tr>
<td>Second Interview</td>
<td>Sales</td>
<td>In Progress</td>
</tr>
<tr>
<td>Contract Proposal</td>
<td></td>
<td>Pending</td>
</tr>
<tr>
<td>Contract Signed</td>
<td></td>
<td>Hired</td>
</tr>
<tr>
<td>Refused</td>
<td></td>
<td>Refused</td>
</tr>
</tbody>
</table>

**Recruiting for a new job**

Tina Robbins has been very busy in her position of Production Manager. It has been decided that there is a need to hire a Production Assistant to assist her in her duties. With the new Recruitment Process application installed, we can now create a new job position and start the recruiting process.

Click on Job Positions under the HR Resources configuration section and click on the Create button.
Here we have filled in the details for our Production Assistant assigned to the Production department. We have entered a short job description and requirements for the job to help in the recruiting process.

We can now click on Launch Recruitment to begin recruiting for this position. This essentially flags the recruiting process as active and makes it easier to determine for which positions we are currently hiring.

**Understanding the Interview button**

The Interview button allows you to select a PDF job survey file to download. This survey could then either be printed and presented to an employee to fill out before an interview or e-mailed to the employee. A separate Survey module can also be installed, which provides a fairly robust form you can use to design your own surveys.

Surveys can be organized around job function and can be valuable for prescreening employees before a formal interview.
Creating an employment application

When a potential employee sends in an application, resume, or another trigger that will allow you to document his/her interest in working for your company, create a recruitment application.

Under the Recruitment | Applications section click on Create to enter a new application.

This form contains many fields, but by default, the only field required for the application is Subject. The rest of the information can be collected throughout the recruiting and interviewing processes.

Most of these fields are self-explanatory. Notice that at the top right, we can see the progress through the various stages for this employee.
Hiring employees
Let us go ahead and hire this Bob Nelson guy. Thankfully, OpenERP’s Recruitment module will create the employee for us by simply clicking on the Hire & Create Employee button.

Looking at the Employees list, we can now see that Bob Nelson is an employee in the Production department with the title Production Assistant.

Working with user access rights and groups
While implementing an ERP system, it is important that you take time to define roles for your users and how they map to the functional job of the employee. The primary reason for including user access rights and groups in a chapter on human resources is that it can be very desirable from a business perspective to have human resources closely involved with both establishing job descriptions and the processes that employees will be working with each day.

Knowing built-in access rights
When you first install OpenERP, the setup process makes certain assumptions about user groups and the various permissions each user has when they are assigned to that group. In Chapter 3, Exploring Customer Relationship Management in OpenERP, we created a sales manager named Mike Zeigler by assigning him to the manager group in sales. This manager group provides Mike with a different level of permissions as compared to a user who is placed in the user group in the sales application.
Implementing the Human Resources Application

So how can we look at exactly what access rights are available to which group?

1. Under the Settings menu navigate to Users | Groups. You will then be presented with a list of the current groups that are configured in your OpenERP instance.

   Here you see the primary list of groups preconfigured when OpenERP is installed.
2. Click on the **Sales / Manager** group to examine the details of this group.

After you have clicked on it, you will be presented with the group details, and the page will default to displaying the list of users that belong to the group.

The form displays all the users that belong to the **Sales / Manager** group along with their login details and primary language.

OpenERP provides quite a bit of flexibility while managing groups. For this reason, it is a good idea to plan how you want to organize your users into groups, including who will be administrators and managers versus common users with limited access capabilities.

**Understanding inherited groups**

Organizing and managing user groups can be quite tedious in any system. OpenERP helps reduce group maintenance issues by offering the ability to inherit groups from other groups.
Clicking on the **Inherited** page allows you to manage from which groups the current group inherits.

Here we can see that any user who is added to the **Sales/Manager** group is also added to the **Sales/ User: Own Leads Only** and **Sales/ User: All Leads** user groups.

While defining your own groups, make sure you take into account the ability to add users to multiple groups and the ability to have groups inherit access rights from other groups. These two abilities, when used together, can dramatically simplify managing access rights.

**Setting access to menus and views**

To the right of the **Inherited** tab in the form, you will find the **Menus** and **Views** tabs. Each of these respectively let you define what menus and views this group will be allowed to utilize.
In this view you can see that we have a variety of menus that are available to the Sales/Manager group. The sequence number is utilized to determine the order of menus for the user. With proper planning, you can develop a menu interface that is most desirable for that specific user group. Menus, submenus, and views that are hidden are inaccessible to a user.

The Views tab has no rows for most user groups. For more complex configurations, views can be defined by groups to get more control over the interface for a given set of users.

To see an example of a view defined by a user, look for the group Properties on lines in the application Technical Settings. This group provides a specialized view for line items on a sales order if you are a member of the Technical Settings group.

Limiting access to menus and views is not the most secure way to prevent access. If denied access to the Products menu under Sales, for example, a user might still be able to access products via the Warehouse or Manufacturing menu. A tech-savvy user might also gain access by guessing the URLs to type into the browser's web address. A much more secure way to prevent access would be to deny access to the Product object under the Access Rights tab.
Determining access rights

For the most part, the Access Rights tab is where you will establish most of the specific permissions for a user. Clicking on this tab will list all the models in the system to which a user will have access to if they are a member of that group.
The preceding screenshot shows a listing of all the default objects (models) and the read, write, create, and delete permissions assigned to the Sales/Manager group.

As a real-world example, let’s say that while we want our sales manager to be able to view products and make changes to those products, we perhaps do not want sales managers to create new products or delete products. In this case, we need to have another control in place, and we would edit these access rights to remove the Create Access and Delete Access permissions for the Product, Product Template, and Product Category objects.

During the deployment of your OpenERP system, take time to understand these access rights for groups in OpenERP. As you add more users to the system, it becomes increasingly important that you don’t simply give everyone full access to everything. By restricting users to only the tasks they need to complete their job requirements, you have better control, and therefore, are likelier to have more accurate information.

**Summary**

In this chapter we examined the various human resource applications available in OpenERP. We installed the base Employee Directory application followed by applications that managed time and attendance as well as leave requests. Finally, we installed a module that allowed us to manage the recruiting processes of new employees. We walked through completing an employment application and finally turning the potential applicant into an employee.

In the next chapter we will look at the Project Management application in OpenERP and how it can be used to improve service quality for customers. It allows you to organize from the most simple projects to complex projects involving multiple tasks. Furthermore, you can even track the time related to projects and display project information in a variety of graphical formats to make it easier to track your deadlines.
In this chapter we will explore a very flexible application that allows you to manage projects and tie them to other applications in OpenERP. The Project Management application allows your company to manage project stages, assign teams, and even track time and job costs related to projects. Analytical accounting features give you even greater control of how project costs can be tied to your company’s general ledger.

This chapter covers the following subjects:

- Discovering various uses for Project Management
- Linking projects with customer accounts
- Assigning teams to projects
- Creating custom project stages
- Adding, assigning, and organizing tasks
- Tying to analytical accounting and employee timesheets

The basics of Project Management

Depending on your industry and the types of projects you may encounter, the Project Management application can be set up to manage independent projects or instead it can be configured to manage projects related to customers or sales orders. With additional modules, it is possible to tie the Project Management application into virtually any aspect of OpenERP.
Understanding Project Management

For example, you could simply use the Project Management application to track the various stages and tasks involved with a company event. Who is going to be responsible for finding the location? When will you need to order the invitations? Who is going to set the agenda? When is an employee going to go and pick up the sound system? In this instance, the Project Management application is simply being used to track a single project that is not associated with the customer.

In other instances, you may wish to utilize the Project Management application to track projects that are organized around your customer records. A common example would be a construction firm. After assigning the project to a customer, you can track various stages of the project life cycle. Employees can be assigned tasks, and using the OpenERP messaging system, you can share project details with your customers. It is in this configuration that the OpenERP Project Management application can add real value to an OpenERP installation and provide better integration with your accounting system with less effort than a standalone project management tool.

Installing the Project Management application

To access the project managing features, you will need to install the Project Management application.

Go to the Settings menu and install the Project Management application using the same process as the previous OpenERP applications.

A real-world project example

Like in other chapters we will use a real-world example to demonstrate the functionality of OpenERP's Project Management application. In the silk screen industry, it can be common to have extremely large projects that can span across many types of apparel and print designs. For this example, we are going to create a project to manage creating an entire line of sports jerseys for an organization called Lil League.
When defining our project, it is important to look at the scope of our project and why it will be valuable to use the project manager to organize the various tasks involved. With our Lil League organization, we have multiple teams that can vary in logo design for the team, the number of players, the sizes of the apparel required, and the printing of different players' numbers and even players' names. There are often multiple deadlines to manage and a number of people that may need to approve various phases of the project as they are complete. Using the Project Management application, we can better track this information and tie it to sales orders and other OpenERP functions.

**Creating our first project**

After the Project Management application has been successfully installed, we can go to the Project application and create a new project.

Under the Project application, select Projects in the menu on the left and then click on the Create button.

![Create Project](image)

After you click on Create, the project editor will allow you to enter various details about your new project. For our example, we will start out defining the name of the project, Sunny Hill Lil League, and assign the project to a customer we have created, Sunny Hill Sports.
Assigning a customer is optional, but in this example we are identifying the customer as the primary league contact.

You will notice that the project is set to In Progress in the top-right corner. You will also notice that you can assign someone as the project manager as well, though this is not required.

If the Tasks checkbox is selected, you will see the Project Stages tab in the form. Different projects can naturally have different project stages that they go through on their way toward completion. For our example, we will leave the Tasks checkbox selected. A little later we will go through how to define the various stages for our project.

Assigning project teams
The first tab, labeled Team, is where you add team members to the project. Click on the Add button, and you will be provided with a list of the available team members in your company. Type the name into the box in the upper right to filter the list or to locate a particular employee by his/her name.
Using the checkboxes to the left of a person's name, you can add multiple project members to the project team at the same time. To mark everyone who is included in the list, use the checkbox at the very top in the header row.

After you have selected all of the project members you wish to add, click on the Select button to add them to the project team. Next to each member in the project team in the project edit screen, you will see a small x. Clicking on it will remove that team member from the project.

**Setting other project info**

In the second tab, we can record additional information related to the project. We can set a start date as well as an end date for the project.

The Parent field provides the capability to nest multiple projects inside another project. If, for example, we wanted to treat each Lil League team as a separate project, we can group them under this project using the Parent field in this form. In this way, an entire smaller project can itself become a mere step (or a task) along the way to completing the larger, more complex parent project. We can examine all the projects in OpenERP, as they are ultimately contained in the Projects collection.
Understanding project stages

In the third tab, we can define the stages for the project. This tab will only be available if the Tasks checkbox in the upper-left corner is selected.

The Project Management application has a set of default project stages that will be automatically populated when you create a new project. You can then add, edit, and delete them as required to meet the needs of each specific project.

In the Project Stages tab you have both Stage Name and Related Status. In our example, you can see that when the project is in the Specification stage, the related status is considered as Pending. When you move into the Design, Development, or Testing stage, then the status is considered as In Progress.

The purpose of this design is so that you can have many unique stages that will then directly set the status of the project. This allows for more flexible and comprehensive project management.

Defining project stages for a specific project

For the purposes of our example, there are several changes we would like to make to our project stages. While Specification and Design are very appropriate stages for our real-world example project of printing team jerseys, the stages of Development and Testing could be much better defined for our example. Instead of Development, the stage could better be described perhaps as Manufacturing; instead of Testing, the stage could better be described as Quality Assurance.
While the stages we may wish to change will depend on the project, for the silk screen printing industry we can anticipate that we will most often wish to use these new definitions for nearly every project. So let us redefine them once so that the new stage descriptions may be used in all future projects.

To edit a project stage, simply click on the row you wish to change. In this case, we will begin by changing the name of the Development stage.

![Open: Tasks Stages](image)

In this example, we have simply changed the stage name from Development to Manufacturing, which better describes the stage for our project. We have also kept the Default for New Projects checkbox selected because we can anticipate that most of the projects for this company will involve manufacturing. The Related Status option can be set in the pop-up menu in the top-right corner. For our purposes, we want to keep Related Status to In Progress. If needed, Sequence could be changed here to reorder the stages of the project.

The Folded by Default checkbox tells OpenERP how to handle stages that have no tasks assigned to them in the project view. If the Folded by Default checkbox is selected, the kanban view will make this stage invisible when there are no tasks currently in this stage. If the checkbox is not selected, OpenERP will show the stage in the kanban view even if there are no tasks currently assigned to this stage. We will keep this checkbox unmarked.
The following screenshot shows the new stage list after it has been modified with stage names more appropriate to the Lil League project for our example:

In addition to changing Development to Manufacturing and Testing to Quality Assurance, we have also added stages for Purchasing and Product Delivery. With these stages in place, we can now assign tasks that will help manage us the project through the various stages.

The little bullet to the left of the names in the Stage Name list will allow you to drag-and-drop stages to reorder them in the list; this is much quicker than editing the Sequence field of each stage. Clicking on the little trash can icon in the far right of the row will remove a stage from the list.

Now with our project stages defined, we can begin defining tasks for our project.

**Defining project tasks**

The main unit for tracking the various activities involved with a project is a project task. We will create a new task by going under the Task menu in the Project application and clicking on the Create button. Here we'll define various aspects of the task:

- Name of the task (this is required)
- Stage to which the task belongs (also required)
- The project to which the task is assigned
• The deadline date of the task
• The responsible party assigned to the task
• Any tags you would like to associate with this task
• A description of the task

For our example we have filled out the task as seen in the following screenshot:

At the top of the form, you will see all the project stages with the current stage highlighted in blue. In this example, Specification is the currently selected stage. When in edit mode, you can click on these stages to directly assign the task to a given stage. This can be changed as the project progresses so that you are not locked into keeping a task assigned to the same stage throughout the project.

The Tags field can be valuable for better tracking and organizing of tasks. In our example, we have defined an information collection tag. This tag can then be assigned to any task that is related to collecting data regarding the project.

**Defining additional task information**

The Project Management application also allows you to define additional information for project tasks. This information includes:

• The priority of the task
• The sequence
• The customer
• The start date and end date in the Gantt view

In our example, we have used the same customer throughout the entire project. For complex projects, you may have several customers that are part of various tasks in the project.

**Creating additional tasks**
For our real-world example we are going to define several tasks at various stages. These tasks include:

• Approving the logo designs
• Collecting the names and numbers of the players in each team
• Identifying suppliers for the product
• Estimating costs and pricing for the raw product
• Placing purchase orders for the raw product
• Creating printed prototype jerseys for customer approval
• Scheduling the print jobs
• Printing the jerseys
• Inspecting each jersey for defects
• Packaging and shipping the jerseys
After entering our tasks and assigning them to the various stages, we can look at the tasks most easily in the kanban view. In this view, you can drag-and-drop tasks to move them to different stages and reorder the sequence of tasks.

Each task is represented as a card and provides the name of the task, the project, and the due date for the task. Any tags, such as information collection, are also displayed on the card.

If you mouse over the little icon shaped like a head, you will see the person assigned to the task and who is, therefore, responsible for it.

The small arrow in the upper-right corner of each kanban card will bring up a pop-up menu that will allow you to specify the color of the card to visually organize your tasks.
Understanding Project Management

The prior screenshot shows how to change the color of the kanban card by clicking on an available color from the palette.

You may also want to see the tasks in both a list view and a Gantt chart view. The icon for the Gantt chart view looks like this: 

Tasks are shown in a Gantt chart view format in the previous illustration. Using this view, you can adjust the time of a given task by clicking-and-dragging the edges of the taskbar. You can also click in the center of a bar and drag a task left or right to reposition it in the Gantt chart view. You cannot, however, drag tasks up or down to reorder them.

If you move your mouse over a task and hover there, you will see a small pop-up box providing you with additional information about the task. This is demonstrated in the following screenshot:
Don't waste your time trying to double-click or right-click on a task in the Gantt view. Unfortunately, OpenERP does not provide a direct way to edit tasks from the Gantt view.

In addition to the Gantt view of the tasks, you can also pull up the tasks in calendar view mode. While in calendar view, on the right you can see the color legend showing each person assigned to tasks.

Fortunately, unlike in the Gantt view, in the calendar view you can double-click on a task to bring it up for editing. You can also use the standard drag-and-drop features of the calendar to set a new due date for the task.
Completing project tasks

As you complete project tasks, you can bring them up and click on the Done button. At this point, the task falls out of the currently assigned stage and is moved to the Done stage. One easy way to see which tasks have been completed is to look at the tasks in a list view. In this view, you will see a green progress bar as well as Done written in the Stage column for every task that has been marked as completed.

As you can see in the screenshot, we have marked off six of the tasks as complete or Done.

If, for some reason, you accidentally mark a task as done or you find out later there is additional work needed for a task, you can edit the task and click the Reactivate button. The task will then return to its previous stage and will no longer be considered done.

Estimating project costs and time

If you have Analytical Accounting activated and the Timesheets human resources application installed, you might also want to install the application called Bill Time on Sheets. With it you can use the Project Management application in conjunction with data collected from timesheet records to track and summarize employees' time and the labor costs related to a particular project. Creating tasks within a project can create lines on timesheets, and vice versa.
Summary

In this chapter we examined the Project Management application. We created an example of a real-world project involving our Lil League organization. After setting up our project and assigning team members, we defined the various stages that would be involved in completing the project. With the stages defined, we were able to go through and assign various tasks to the stages along with their dates of completion. Finally, we looked at the various ways you can view the tasks and how you can complete them.

In the next chapter we will explore how you can create advanced searches and custom dashboards in OpenERP. As a company uses its system day to day, the amount of data collected can grow quite rapidly. Being able to locate pertinent records in a speedy fashion is vital for optimum business operation. We'll discover how to utilize all the handy searching, filtering, and dashboard presentation tools that are at our disposal within OpenERP.
Creating Advanced Searches and Dashboards

In this chapter we will cover advanced searching, custom filters, and dashboards. We begin by looking at how OpenERP searches the various data sets within the system. Next, we will explore more advanced searching options and discuss how you can save these filters so they can be easily accessed when you need them. Finally, we discuss the OpenERP dashboard capabilities and how we can improve usability for users.

Topics covered in this chapter include the following:

- Identifying users' search requirements
- Understanding default filters versus custom filters
- Grouping items in a list
- Setting and saving advanced search conditions
- Creating dashboard content and layouts

Determining the search requirements for your business

One of the tasks that can often be frustrating and time consuming for users is trying to find the information they need. When data sets are small and simple, there is not much of an issue. As the number of records in the system grows, it can become increasingly hard to find information.
Creating Advanced Searches and Dashboards

When implementing an ERP system, you will want to take the time to work with users and get familiar with the data they use each day. If you are working with a purchasing system that only produces an average of 10 purchase orders a day, you will have far less concern over advanced searching in that application. However, if you have 20 purchasing agents cutting 450 purchase orders a day, it will be critical that the users have a firm grasp on the search functionality of the system. Trying to locate a particular order can be like trying to find a needle in a haystack.

Take the time to sit with users and watch them use the system. Often users will need to look up the same types of data repeatedly in their daily interaction with the system. These are the activities for which you will want to set up custom filters and perhaps even include those filters on the users’ dashboards.

Fortunately, OpenERP offers a robust searching mechanism as well as the ability to create dashboards to display information that the user may need to look at frequently.

For the purposes of this chapter, we have created a new database with the demonstration data so you can better see the searches in action.

**Searching in OpenERP**

OpenERP provides a standardized search box in the top-right in all of the list and kanban views. Depending on the menu item, some forms come with predefined filters already set for the list.

If you navigate to Sales and select Customer, you can see the search box in the upper-right corner with the Customers filter preassigned.

Some lists open with a predefined filter that will limit the primary data set. In the prior screenshot, you can see that the list view Customers has a customer filter applied by default when you open the form. OpenERP stores customers vendors and employee information in the same central database table. The Customers filter prevents other types of data such as vendors or employees from displaying in the list.
In this instance, if you clear the Customers filter by clicking on the small close box in the tag, you will have a list with not just customers. Instead, OpenERP will return partners, users, suppliers, and contacts as well.

Basic searches are handled easily in OpenERP. Just go into the search box, begin typing, and press the Enter key. OpenERP will then look at the primary search fields for the type of data you are searching and show you the results in the list or kanban view.

In the following screenshot you can see a simple search:

In this example, OpenERP has returned all the customers that have Bank in their name.

Now we can see that there are two filters applied: the default filter Customers that was there when we opened the customer list and the Name filter that will limit those customers to just the names that include Bank.
The small space between the two filter tags means that both conditions are required for a record to be included in the results list (A and B).

When two filter tags are butted up against one another without a space between them, this denotes records may meet either condition (A or B).

OpenERP will remember your search criteria as you move between list, kanban, and form views. Once you go to another menu item, the search criteria will reset to the default search when you return.

The small magnifying glass in the far left of the search box will allow you to repeat your search again. This would be most useful in an environment in which transactions are coming in quickly and you want to refresh your results with the latest data.

The preceding screenshot shows the search again magnifying glass to the left of the Customers filter.

As you type in the search box, before hitting the Enter key, OpenERP will display the available results in a small drop-down list directly under the search box. With each letter you type, the results list will narrow. OpenERP displays at the top of this list which of the primary fields it is using to conduct the search.
Type the letter R into the search box. Notice in the previous screenshot that OpenERP is searching the Name field for R but allows you the option to search other fields like Tag, Salesperson, and Related Company. You can use your mouse cursor or arrow keys to search or filter by one of these other fields; however, if you simply press the Enter key, only the Name field for customers will be searched.

**Using filters in list views**

OpenERP provides default filters for all of the list views. Applying a filter will limit the records that OpenERP is displaying. You can apply one or more filters depending on your needs. The available filters and by what fields you can group will vary depending on the data you are viewing.

For example, the Products view will have a completely different set of filters and group options than the Customers view.

While each search box will have different default Filters and Group by... options, the functionality is the same. There is an area at the top that contains the default Filters and Group By... options. The second section is the Custom Filters area, where all of the user-defined filters will be stored for later use.
Creating Advanced Searches and Dashboards

Next, there is the **Advanced Search** section and finally an **Add to Dashboard** option at the very bottom to append the current search criteria to a dashboard.

In the example drop-down menu in the preceding screenshot, you can see that we have checkmarks next to the filters that have been applied. The tags for the filter are also displayed inside the search box. With these two filters applied, OpenERP will show the contacts or persons who are also customers. Clicking on a filter applies the filter immediately and refreshes the result list.

Naturally, the list of available filters will change depending on which set of records you are viewing. Clicking on a filter that has a checkmark will remove that filter from the search.
Grouping information in lists

In addition to filtering your lists, you can also group data in most list views using the Group by... option. When you group data in a list, a little triangle appears to the left of each group header. Clicking on this triangle will display the rows grouped under that header.

[Table showing grouped data]

Filters and groups can be combined together to produce a list of results. To clear all search conditions and groupings at once, click on the circled x on the far right of the search box. As shown in the previous screenshot, we have filtered by products that can be sold by selecting the Can be Sold option in the Filters menu; we also grouped our data by category by selecting Category in the Group by... menu on the right. Next, we expanded the Accessories category by clicking on the small triangle to the left. You can then see the list of products that are included under the Accessories category. As with filters, clicking on Category again will remove the grouping. You can also nest groups inside of other groups simply by selecting additional items under Group by...
Creating Advanced Searches and Dashboards

Grouping can be a great way to look at data. Unfortunately, with extremely large data sets, grouping lists can be very slow because far more records must be processed if you are filtering and browsing data.

When you group records in kanban view, the results are presented in columns rather than in rows. If there are many columns, you can use your scroll bar at the bottom of the window to browse the data.

The preceding screenshot shows Products in kanban view, demonstrating how a user will need to scroll not only up and down but also left and right to get a view of all of the items.

You need to be somewhat careful when grouping in kanban view. If you group on a field that has many results, you will have a long way to scroll to the right to even turn off the filter. In this case, it is usually faster just to click over to another menu and come back to restore the default search settings.
Performing an advanced search

While the default filters may help us find most of the data records we seek, it is inevitable that there will come a time when we will need a more customized search. To create an advanced search, click on the down arrow in the right of the search box and then click on **Advanced Search** to expand the available options. Here we will get a drop-down list of fields that can be used to set our search criteria.

Choosing a field from the list will allow you to select from the available search operators as well as specify the data for which you wish to search. Click on **Add a condition** to enter further criteria. Clicking on the small x to the right of a search condition will delete that condition from your **Advanced Search**.

In OpenERP, you will often find it a best practice to make records inactive when they are no longer required. For example, if you discontinue a product, you will often find yourself unable to delete that product because there are transactions tied to it. Therefore, you will wish to inactivate that product record. By default, OpenERP will hide inactive records. If you need to retrieve inactive records, use **Advanced Search** to create a condition where the **Active** field is **false** and then apply it to this filter.
You can continue to add additional criteria to your Advanced Search. When you have specified all the criteria you wish to use in your search, click on the Apply button to apply the custom filter.

In the Advanced Search option, we have specified two conditions: Credit Limit must be greater than 550 or the Date that the customer was acquired must be greater than 07/11/2013. Many users can get confused and believe that this filter would imply both Credit Limit and Date must meet our criteria, but this is not so. Notice the faint little or to the left of the second condition.

**Specifying multiple advanced searches**

As you can see in our previous example, OpenERP will always use an or operation between each of the conditions you add to the search. But what if we wanted to have a search where the credit limit is greater than 550 and the date the customer was acquired is greater than 07/11/2013? To accomplish this, you must first apply the advanced search with only the credit limit condition defined. That will limit the results to only records of customers that have a Credit Limit greater than "550". Then, you can go back and add a second advanced search that only contains the Date greater than "07/11/2013" condition.

Just remember that if you want both conditions to be true then they must be applied separately. If you want either of the conditions to be true, then add them together in one search.

In the previous screenshot, we created a search that will return customers that have a credit limit greater than 550 or a customer acquisition date greater than 07/11/2013, and each of those customers must also have total receivables less than 1000.
Saving your advanced searches
While advanced searches are quite powerful, they can often take a bit of time to configure and get the results just like you want them. Fortunately, OpenERP allows you to save your searches so that you will not have to build them from scratch each time. To save a custom search, click on the little triangle next to **Save current filter**, provide a name for the search, and then click on **Save**.
Once you click on **Save**, the filter is added to your list of **Custom Filters** and can be applied just like the default OpenERP filters. In addition, you also have the option to save the custom filter for all users and even set a custom filter as the default filter to be applied when you bring up the list.

In the preceding screenshot, we applied the custom filter, **Credit Customers**, that we just saved. As you can see, the criteria at the top no longer shows all the detail in the advanced search, and instead uses the name you provided when you saved the custom filter.

There is not an easy way for an end user to see what the criteria of their search is after they have named and saved their search for later use. Like in our example, Credit Customers is all we will see when returning to the search later. Until OpenERP provides an easier method, users should be encouraged to document their searches.

The ability to save advanced searches into your own custom filters and make them available for other users allows you to better customize OpenERP for your business requirements.

**Adding information to your custom dashboard**

Dashboards allow you to take information that you need to look at frequently and put it together in one place. OpenERP has a very flexible dashboard system. Each user has a personal dashboard named **My Dashboard** provided with the default OpenERP setup.
To add a new result set to your dashboard, simply click on the little triangle next to **Add to Dashboard**. By default, OpenERP will prompt you to add the search list to your own personal dashboard. However, if you wish, you can add the results to any dashboard by selecting the name of that dashboard in the list and clicking on the **Add** button.

In this example, the current **Credit Customers** filter we created in the previous step will be added to **My Dashboard**, which can be found as the first option under the **Reporting** menu.

OpenERP provides a variety of layouts, so you can customize the appearance of the dashboard according to your preference. For example, you may wish to have two columns of lists summarizing your sales or, if there are view columns, you may choose to have a column of three lists.
Creating Advanced Searches and Dashboards

Clicking on the **Change Layout** button will bring up a small pop-up window which allows you to select an alternate layout.

In the top-right corner of each item added to the dashboard, you can click on the little underscore icon to collapse the report area down to just its title. To arrange items on your dashboard, simply click and drag the item to drop it in the desired location. Finally, you can remove an item from the dashboard by clicking on the close box in the upper-right corner of the item.
In this example, we have added a few more items to the dashboard and arranged them into two columns. Adding a graph is just as easy as adding a list view to the dashboard. In this example, we went under the quotations, changed the view to graph, and then added it to our dashboard. For graphs there is a small pop-up menu on the top-right. It allows you to change the properties of the graph.

From this menu, we can switch between pie, bar, and line graphs. We can also designate where to display the **Legend**, if at all. We can also swap x axis and y axis, display the raw data, or save the graph as a picture file on our local computer.
Summary

In this chapter we examined OpenERP’s advanced searches and dashboards. Advanced searching allows you to search on a variety of fields as well as save your searches so you can easily pull them up later. Using these features, you can more easily find the data you are looking for and place data that you need frequently into your own personal dashboard.

In the next chapter, we will begin to take a look at how you can customize OpenERP to meet the specific needs of your organization. We will discover how to activate developer mode, which will allow us to append fields to OpenERP's screens. We'll then begin adding our own fields to forms for collecting data, to lists for finding and displaying the data, and to models which handle all the rules and methods for storing the data to the underlying PostgreSQL database. But first, what is the single most important thing that we need to do before customizing our system? That's right, we need to know how to back up and restore our data, just in case.
Customizing OpenERP for Your Business

In this chapter we will begin covering one of the greatest advantages of OpenERP: the ability to customize the software to meet the unique needs of your business. Fortunately, OpenERP provides a great deal of flexibility in which you can customize OpenERP without writing any code or developing a module. We will begin by learning how to activate the OpenERP developer mode and then back up our database. This is a very important practice when customizing OpenERP. After that, we will learn how to add fields to our database and display them on forms and views.

The following topics are covered in this chapter:

- Getting in and out of the developer mode
- Backing up our database
- Restoring data from a backup
- Appending custom fields to models
- Displaying our newly added fields upon forms and list views

Activating the developer mode

To customize OpenERP, the first step is to activate the developer mode. Once you enter this mode, OpenERP will provide you with a lot more onscreen information as you navigate through the interface. This mode also allows you to make changes to the database and store that information in a file.
To activate the developer mode, go to the menu in the top-right corner of the screen. This menu will have the label of the user you are currently logged in as. In our case, this is the Administrator menu. Choose About OpenERP from the menu.

Once you have entered this screen, you can click on **Activate the developer mode** to begin customizing OpenERP.

OpenERP recognizes that you are in the developer mode by adding &debug=# to the URL in your web browser. Additionally, OpenERP changes the information that is provided when you move your mouse over various fields in the interface. For example, when viewing a sales order record while in the developer mode, you can move the cursor over the **Date** field to reveal details about how that field is represented internally in OpenERP.
The preceding screenshot demonstrates how OpenERP displays information while you are in the developer mode. In this example, we can see that the Date field is named date_order and that the field belongs to the sale.order object. Additionally, we can see, the field type is date and there are modifiers assigned to the field. This information will be of great value as you continue to customize OpenERP.

**Getting out of the developer mode**

Now that you are in the developer mode, there will come a time when you want to exit developer mode and work with OpenERP as you normally would. To exit developer mode, simply remove debug=# from the URL string in your browser.
Make sure you leave the & symbol in place when you remove the debug tag from the URL. If you get any errors or other unusual behavior after removing debug=# from your URL, you can typically use your browser's back button. If this also fails, you can always restart the browser and log back in to OpenERP.

**Backing up your database**

When you make changes while in the developer mode, those changes are written into the database associated with the company. One of the major advantages to this approach is that you do not have to write code in Python or create a custom module to implement simple customizations. One of the major disadvantages, however, is that there is the potential that you could make a change that is undesirable and perhaps difficult to reverse.

Therefore, it is very important and highly recommended that you make backups of your database both before and after you make any customization through the developer mode.

---

**Do not skip this step!**

It is always a good idea to frequently back up your database. However, it is absolutely imperative that you back up your database before undertaking any customization.

---

To back up your database, you must first log out of OpenERP. After you have successfully logged out, click on the **Manage Databases** link on the login form. From this screen, you click on the **Backup** link in the top-left corner of the **Database Management** menu.
To back up your database, select **Database** from the pop up menu and enter the **Master Password** for the OpenERP installation (by default, **Master Password** will be `admin`). Next, click on the **Backup** button.

After you click on the **Backup** button, OpenERP will then save your database to your local drive. Depending on the browser you are using and the settings in your browser, the prompt you get to save your file will vary. The default filename will end with the `.dump` extension.

After you save your file, OpenERP will download it into the directory you have specified. If this is the first time you have backed up your database, you should also take the time to verify that you can successfully restore the database. While this may seem like an unnecessary exercise, it is important to remember that a backup is only as good as your ability to successfully restore it.

### Restoring a database in OpenERP

To restore a database in OpenERP, click on the **Restore** option in the **Database Management** menu.
Customizing OpenERP for Your Business

To restore your database, you need to provide three pieces of information: the backup **File** you wish to restore, **Master Password** of OpenERP, and **New database name**. Clicking on the **Browse...** button in the file selection area will prompt you to select the `.dump` file created when you performed the backup.

![File Upload](image)

After you have specified the file and the other required fields, click on **Restore** to begin restoring the database. A small progress bar in the bottom left of the browser will update you on the progress of the restore. Once the restore is complete, log in to the database to make sure everything is as expected.

Now, with a successful backup and restore, you are ready to begin customizing OpenERP. If something goes wrong, you will have the ability to restore your backup. While customizing OpenERP, remember to back up the database frequently and test the restore process often.

**Adding a custom field to OpenERP**

One of the most common reasons for customizing OpenERP is to collect additional information that is specific to your company. If you are running an insurance company, perhaps you want to specify the policy number on your sales order. If you are working in property management, perhaps you would like to store the date in which the lease agreement will expire.
For our working example, we will be adding fields that will help us better manage the data and processes for our silk screen company. Specifically, we will be adding the following fields to the sales order header:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Label</th>
<th>Field type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_daterequired</td>
<td>Date Required</td>
<td>Date</td>
<td>In the screen printing industry, deadlines drive when production must begin and when the product should be delivered to the customer.</td>
</tr>
<tr>
<td>x_rush</td>
<td>Rush Order</td>
<td>Boolean</td>
<td>If related to Date Required, it is necessary to flag some sales orders as rush orders. Rush Order can then be prioritized and given expedited treatment.</td>
</tr>
</tbody>
</table>

Custom field names in OpenERP should be preceded by x_. This is so that field names in future OpenERP versions and standard updates will not accidentally conflict with the custom fields you have added.

The easiest way to add a custom field to a form is by using a menu that appears on that form. In this case, we will be adding the fields to the Sales Order form. To begin adding a field, navigate to Sales | Sales Orders, open an order in form view and choose Manage Views from the developer's pop-up menu in the top-left corner of the form. The developer menu does not have a label but displays Debug View# as its first option.
By default, OpenERP will select the view that you were on when you chose the Manage Views option from the menu. In this case, the view currently selected is the sale.order.form view. For now, ignore other values such as External ID and Inherited View. Click on the Edit button to modify the sale.order.form view.

After clicking on Edit, you are presented with the View Editor. This form allows you to add, delete, and modify the fields that are displayed in the view. At first, this form can be a bit overwhelming in its complexity, but it will get easier after using it a few times. In the wide column on the left, you will see the field names of the view organized in a hierarchical tree format. By default, all of the tags and fields of the view are expanded, but you can use the small triangles to the far left to collapse the levels.

Scroll down until you find the field name client_order_ref. It is selected in the preceding screenshot labeled View Editor 489 - sale.order.
Now click on the small blue plus (+) icon in the far right set of buttons. This tells OpenERP that you wish to add a field at that specific location in the view.

![Properties form]

After clicking on the plus (+) icon, you are presented with a Properties form. The Node Type field is pre-populated with field, which is the value that we need to add our field to the form. The Position field is set to After, which tells OpenERP that we wish to add our new field after the client_order_ref field on the form view.

If the field we wish to display already existed in the sale.order model, we could simply select it from the pop-up menu next to field and it would display on the form view. Instead, click on the New Field button to add our brand new field to the Sales Order form.

![Field editor form]

Now you will be presented with a field editor form, as depicted in the preceding screenshot.
The x_ prefix is already specified in the Field Name by default to encourage the use of good naming conventions. We have filled in the other data required for the field including setting the Field Name to x_daterequired, the Field Label to Date Required, and the Field Type to date. Additionally, we have clicked on the Required checkbox so that the user will be required to specify this value when entering a sales order. For this example, we will leave the rest of the form as is.

Click on Save to finish adding our new field to the sale.order model.

OpenERP will automatically select x_daterequired in the pop-up list. At this point the field has been added to the sale.order model in the database. We are now ready to add this field to our Sales Order form.

Click on Update to proceed.
In this Properties form, we can specify how we wish the field to appear on the form. For example, we could override the field label we set in the field properties by specifying a different value in the String field on this form.

A Name field may contain a maximum of 64 alphanumeric characters and underscores but no other special characters and no spaces. The name may often appear cryptic, but end users of OpenERP typically do not need to ever see these names. The String field, however, will be used as a label on forms to prompt for user input. A String field can contain up to 256 characters including spaces and special characters.

Here it is also possible to restrict the field to specific user Groups using the list at the bottom. For now we will accept all the defaults and click on Update to close this screen. Next, click on Close on the View Editor and again click on Close on the Manage Views window.

OpenERP will automatically refresh the Sales Orders form, allowing you to see the Date Required field you have just added.
Adding another field to the sales order

Following the same procedure, let's add the **Rush Order** checkbox to the **Sales Order** form. If you get lost, simply refer to the procedure for adding **Date Required** to the form.

The following is a summary of the process:

1. Select **Manage Views** from the pop-up menu in the top-left corner of the form.
2. Click on the **Edit** button to modify the view.
3. Scroll down and click on the + icon to the right of the *x_daterequired* field that we just added.
4. Click on **New Field** on the **Properties** form.
5. Fill out the form using *x_rush* for **Field Name**, **Rush Order** for **Field Label**, and **Boolean** for **Field Type**. Do not check the **Required** field for this field as we want the ability to not have the **Rush Order** option checked.
6. Click on **Save** to save the field.
7. Click on **Update** to tell OpenERP to add the new field to the view.
8. Click on **Update** again to take the default values for the field view.
9. Click on **Close** to close the **View Editor**.
10. Click on **Close** again to close the **Manage Views** form.
If you followed the preceding steps you should now have the **Rush Order** checkbox added to the **Sales Order** form.

Using the same method you can add fields to almost any form within OpenERP.
Adding a field to an existing view

After adding our fields to the sale.order model and displaying them on the Sales Order form, it would be nice to include these fields in our sales order listing. This way users can see at a glance when orders are due and if they are a rush order.

We begin by going to the Sales Order list view and choosing Manage Views. Next, click on Edit to bring up the View Editor for the sales order list.

![View Editor](image)

You will notice that this list is far less complex than the one for the Sales Order form view. This is simply because a list view has much fewer options and fields on it than a form which edits and manages the data for just one record at a time.

Click on the + to the right of the date_order field. This will pop the Properties window.
After selecting `x_daterequired` in the list, click on Update twice. Then you can follow the same process to add the `x_rush` field to the view. Finally click on Close twice to shut the View Editor and the Manage Views form.

The screen will refresh and you will see our two new fields in the Sales Order list view.

Summary

In this chapter we started by learning how to activate OpenERP developer mode. Next we walked through how to make a backup of the OpenERP database and how to restore that database using the manage database tools in OpenERP. It was emphasized how important it is to backup databases prior to performing any customization in OpenERP. Next we went through step-by-step how to customize OpenERP by adding fields to the database and ultimately to forms and views.

In the next chapter, we will explore how we can use the OpenERP workflow editor to customize OpenERP to adapt to the demands of your business processes. For our example, we will be extending the standard OpenERP workflow so that sales orders can only be confirmed once the artwork for that order has been approved.
Understanding Workflows

One of the key objectives of an ERP system is to organize information as it passes through business processes in the company. As operations grow in complexity, companies typically attempt to better define their processes by diagraming and creating formal workflows. While there are a variety of workflow methodologies, the primary goal is the same: to provide clear documentation of the processes that make up the core operations of the business. The better these processes are defined and executed, the better the company can manage their operations.

In this chapter, we will:

• Take a look at OpenERP’s business-friendly view of a workflow
• Use the OpenERP graphical workflow editor to examine the sales order workflow
• Study how activities and transitions can be used to manage the state of a document in OpenERP
• Learn how to modify workflows with a real-world example: requiring an art-approval process before the sales order can be confirmed

OpenERP facilitates the ability to map business processes to OpenERP by providing a workflow designer. This designer is utilized for many of the major OpenERP applications including sales, purchasing, and invoicing. OpenERP provides both a high-level view of the workflow processes as well as a developer mode that allows an experienced OpenERP developer the ability to alter the default workflows in OpenERP.

Modifying workflows should only be attempted by experienced OpenERP developers. While OpenERP offers a visual interface for modifying workflows, there is a real risk you could break existing workflows. Before modifying any workflows, make sure it is on a non-production version of OpenERP and you have a viable backup.
Example of OpenERP workflow

When you have been putting in quotes and converting them to sales orders, behind the scenes, OpenERP has been following a default workflow for that process. Let's take a look at the OpenERP default sales order workflow.

Activating the developer mode

Like in the previous chapter, we will need to activate the developer mode to view the workflows in OpenERP.

Choose About OpenERP from the menu in the top-right corner and click on Activate the developer mode. For more detailed instructions, see the Activating developer mode section of Chapter 10, Customizing OpenERP for Your Business.

To view the sales order workflow process, go to the Sales menu and choose Sales Orders. On the right, next to the Sales Orders label at the top of the form, you will see a small question mark icon. Click on this icon to view the sales order workflow process.
The following screenshot is a high-level overview of the sales order process:

This workflow is considered as the non-developer version. While not necessarily intended for end users, this workflow could be used to better illustrate system functionality to team members.

Each rectangle symbol in the figure is called a node and contains a step in the workflow process. The arrows indicate the flow of information through the workflow. Each node contains a short description to summarize that step in the process. In OpenERP, as your document moves from Draft to Confirmed to Done, it is the arrangement and organization of these workflow steps that determine the document stage.

Nodes that have a light circle symbol in the center contain subworkflows.

For example, the invoice node in the preceding figure will have its own workflow that determines how invoices are processed.
Understanding Workflows

Viewing the actual step in the workflow
OpenERP provides an easy way to jump to the corresponding document or list of documents from a node in the workflow. Simply click on the little green arrow in the lower-right corner and you will be taken to the OpenERP model that is associated with that node. In our Invoice node example, clicking on the green arrow will take you to the Customer Invoices list view.

Getting help from OpenERP
Another really nice feature of this workflow figure is that you can click on the blue information icon in the lower-left corner of any node to bring up the associated OpenERP help on their website. Clicking on the icon in this example will take you to the Invoices help section at doc.openerp.com.

Planning the changes to our workflow
One of the most common changes in workflow is to create an additional step required in a process. For example, a company may wish to have an extra layer of approval if a purchase order is over a certain amount. Another example would be a quality assurance step that would be performed before a delivery order can be shipped. For our real world example, we are going to require that a customer must approve the artwork for an order before it can be put into production. However, before we can begin modifying the workflow, it is important that we take the time to look at the standard sales order workflow in OpenERP.

Looking at an OpenERP workflow in detail
The workflow that we have been currently studying is a summarized view of the actual workflow that OpenERP utilizes to carry out the different processes. The developer view of the workflow describes the conditions to determine the various document states and how each process should be handled. To understand a workflow, it is important to recognize exactly what a workflow can accomplish in OpenERP.
In OpenERP, a workflow can:

- Modify the states or stages that a document is currently in
- Determine what conditions or states in a document are required to execute a process
- Make process decisions based on user roles and access privileges
- Create subworkflows to encapsulate complexity

OpenERP also provides a very nice visual workflow designer that allows you to both analyze workflows as well as make some limited modifications. I am specifically stating limited modifications as you will likely need to have at least some experience in OpenERP development if you are going to make anything but minor changes to workflows.

**Looking at the Sales Orders workflow inside the workflow designer**

To access the visual workflow designer for the Sales Orders workflow, go to the Sales screen and choose Edit Workflow from the developer Debug pop-up menu in the top-left of the form. In the following screenshot, we are selecting Edit Workflow from the developer menu.
After you have clicked on **Edit Workflow**, OpenERP will bring up the workflow in a list format. We want to look at the visual figure of the workflow. Click on the **Diagram view** icon on the far right to view the workflow visually.

Once you have selected the diagram view, you will be presented with a screen that will look similar to the workflow in the following screenshot:

At first, this sales order workflow is going to look very intimidating. Honestly, it is a rather intimidating workflow altogether. While OpenERP certainly makes it possible to modify workflows, this is often best left until after you understand OpenERP workflows thoroughly. This chapter will help get you started, but be prepared to spend a great deal of time studying OpenERP workflows if you intend to make extensive modifications with any degree of success.
Your first goal should be to gain a solid understanding of how the workflow functions before attempting to make any changes to it. Also, it cannot be stressed often enough: make sure you backup your database before making any changes to an OpenERP workflow.

Workflow nodes
In the developer view of the workflow, the oval symbols are the workflow nodes. Each node can act on documents and execute processes. Nodes that are shaded in grey indicate that node begins or ends the workflow. For example, draft is a beginning workflow node. This, of course, corresponds to the state in which sales orders start when they are first created.

Looking closer at the sales order workflow
We are going to examine in depth just a small part of the sales order workflow. Primarily, we will study the initial nodes where we create a draft sales order and then either send, cancel, or confirm that sales order. The following screenshot focuses on the initial steps of the sales order process:
Take a look at the **draft** node. We can see from the preceding figure that there are three potential pathways out of the **draft** node in this workflow. One of the arrows goes to the **sent** node, another to the **cancel** node, and finally one goes to the **router** node. The option to send, cancel, or confirm a sales order directly acts on the **draft** node in the workflow.

A quick look at the options we have when we create a sales order helps us demonstrate how these functions map to the workflow.

The following screenshot shows the commands that are available when our sales order is in a **draft** state:

- **Send by Email**: This button will trigger the `quotation_sent` and proceed to the **sent** node
- **Confirm Sale**: This button will trigger `order_confirm` and proceed to the **router** node
- **Cancel Quotation**: This button will proceed to the **cancel** node

A question or two may now come to mind. Why are there only three available transitions out of the **draft** node? What about **Save**? What about **Print**? Workflows primarily determine the state of the document. In this case, OpenERP doesn't care to have a printed process in the workflow. You can print the document; but it has no effect whatsoever on the workflow. You can also **Save** the document, but saving that document does not inherently impact OpenERP's workflow. However, when we actually send a document by e-mail, OpenERP has implemented a workflow process (`quotation_sent`) that allows us to see the sales order as being sent to the customer.
Examining a workflow node in detail

Now let's further examine the *draft* workflow node by double-clicking on the node. In the following screenshot, we can see the Activity screen for the *draft* state:

![Activity screen for the draft state](image)

Naturally, the name of this node is *draft*. We can also see that it is tied to the Workflow field which is named as *sale.order.basic*. This node's oval was shaded gray because Flow Start is checked. It is the beginning of the sales order workflow. Workflows can only have one start, but they can have multiple stops.
Click on the **Transitions** page to see how this node functions.

The **Destination Activity** column is what ties this node to the other nodes in the workflow figure. So, for example, when you click on the send button to send a quotation, this node receives the *quotation_sent* signal and then routes the sales order document to the *sent* node. If you click on the cancel button, the *cancel* signal is sent to the *cancel* node.

Take a minute to compare the preceding three destination activities to the workflow figure. It is critical you understand why, for example, the sales order will never go to the *sent* node in the workflow if you have not clicked on the send quotation button in the sales order form.
Transitions in workflows

Transitions between the nodes are what literally hold the workflow together. You can see the transition from a different perspective by double-clicking on the transition in the workflow designer. Let's take a look at the cancel transition by double-clicking on the arrow connecting the draft node to the cancel node in the figure.

![Transition Diagram]

The Source Activity is clearly labeled as the draft node and the Destination Activity is clearly labeled as the cancel node. This data is the same data as we viewed in the draft node under the Transitions page.

For this example, we have intentionally picked a very simple transition. If the user clicks on the cancel button, the cancel transition just goes right on to the cancel node in the workflow. The condition is always True, meaning that we are not examining anything in the sales order or requesting any other input from the user. If for example you only wanted certain user groups to be able to cancel a draft sales order, you could assign the group under the Group Required field in the form.
Ending the workflow

Every workflow needs to have a beginning and of course an ending. A document in OpenERP can end in a variety of ways. Most often the conventions are Done or Cancel depending on the outcome of the workflow. Following our simple draft sales order example, we are going to look at the cancel node and see how we define the ending of an OpenERP workflow.

Double-click on the cancel node to bring up the details.

On the right, we can see that Flow Stop is checked. Additionally, the Kind field is set to Stop All. These settings tell OpenERP that this workflow has ended. Many OpenERP nodes are also tied directly to Python actions. In this example, you can see that the action_cancel method is called during this step in the workflow. Combining the flexibility of workflows and Python scripts allow for an experienced OpenERP developer to create customized industry solutions.
Modifying workflows

The first step to modifying workflows is to have a complete understanding of the existing workflow. Next, you need to determine exactly how you want the updated workflow to function. In this chapter, we will make a modification using the visual editor and modifying views through the developer tools.

Changing workflows is not something that is for novice OpenERP users. It is very possible and quite likely that when experimenting with modifying workflows you will break something. Make backups and make sure that you can successfully restore those backups. Only after you fully understand how the exiting workflows function and you have solid backups should you even consider modifying workflows in OpenERP. Never modify a workflow on a live production system.

Adding a step into the workflow for art approval

In our real-world workflow, it is critical that the right artwork is designed and used for the t-shirt order to be accurate. This art approval step is important and we do not want a sales order confirmed and put into production until the artwork for that order has been approved. There is an extra step we wish to add that will make sure a sales order is not confirmed and put into production unless the artwork for that order has been approved.

Editing the sales order workflow

If you are not already in the developer mode, enter it now. Follow the steps from the earlier sections of this chapter to edit the sales order workflow. Navigate to the sales application and pull up any sales order. Click on the Developer menu in the upper-left corner and choose Edit Workflow. Next, click on the small diagram icon on the far right to visually view the sales order workflow.
Understanding Workflows

Now you should see the familiar sales order workflow that we will modify to include a step for art approval.

We will now add an additional activity to the workflow that will require that art is approved before we can process the order.

Once we begin modifying this workflow, the operation of the sales order module will be changed. It is highly recommended that you perform all of these steps on an instance of OpenERP that is disposable.

Summary of the steps for changing the workflow

Modifying a workflow will typically involve several steps. It is important to outline them before we begin.

Goal: Before a sales order may be confirmed, the art must first be approved.

Steps:

1. Add a new activity to the workflow to handle the art approval.
2. Modify the transitions in the workflow to allow for the art approval activity.
3. Add a button to the sales order editor to Approve Art.
4. Modify the Confirm Order button, so it is only active once the art has been approved.
Adding a new activity node to a workflow

We add a new activity node to the workflow by clicking on the **New Node** button located in the upper-left corner of the workflow editor.

![Screenshot of the New Node button in the workflow editor](image)

After you have clicked on the **New Node** button, a form will pop up that will allow you to define the activity for the art approval process.

![Create Activity form](image)
We have named the activity as `approve_art`. This will let us identify it easily in the workflow. The `Workflow` field is already defined for us as `sale.order.basic`. Under `Kind`, we have changed the default setting of `Dummy` to `Function`. With `Function` selected, OpenERP will execute the code provided in the `Python Action` setting at the bottom of the form.

**Modifying the sales order state in the activity**

The entire purpose of this `approve_art` activity is to set the state of the sales order document to `art_approved`. This state will then be a prerequisite to display the `Confirm Order` button on the sales order. After you have filled out the create activity form, make sure you click on `Save` to commit your changes. Next, we will create the transition that will trigger this activity.

**Modifying the transitions to include art approval**

Adding the activity will leave an oval named `approve_art` floating in your workflow. This is because we have not attached the activity to any transitions. Let's begin first by hooking up the `approve_art` activity to the draft node with a transition. This meets our goal. As long as the sales order document is in a `draft` state, we want the ability to trigger the `approve_art` activity.

To create the transition, click on the `draft` node. Then, using one of the four black handles on the node, drag it to the `approve_art` activity.
Defining a transition

We will now define our transition to call the approve_art activity and set the state of our sales order to art_approved. So what is the trigger? A button on the sales order form will provide the necessary signal to call the approve_art activity. We will define this transition as follows:

[source_activity]
[destination_activity]
[signal]
[condition] True
Understanding Workflows

When you drag between the two nodes to create the transition, the source and destination activity are filled in automatically. We then set the **Signal (Button Name)** to `approve_art`. This means that for this workflow to be functional and for the `approve_art` activity to be called, we must have a button named `approve_art` on our order form.

Once we click on the `approve_art` button, the `approve_art` activity will be called. That activity will then set the state of the sales order to `art_approved`.

**Restricting the art approval to the sales manager**

In defining our transition, we have set the group required to `Sales / Manager`. This means that in order for the art to be approved the user must be part of the sales manager group who has communicated with the customer and given the go ahead for the artwork.

Once you have clicked on `Save`, our `approve_art` activity is hooked into the `draft` node. However, we still have not defined where the workflow should go from there. We must now wire the `approve_art` activity into the workflow so that we can confirm the order and continue processing the sales order.
To connect the approve_art activity to the router activity, select the approve_art node then click and drag one of the four dark circle handles and drag to the router activity. You will then get a dialog for you to specify the transition between the two activities.

![Create: Transition dialog](image)

Like in the previous transition, we created the source and the destination is filled in automatically. Our Signal (Button Name) is set to order_confirm. Why order_confirm? Because we are going to change the order_confirm button to only be available once the document is in the state art_approved. When the art has not been approved, the Confirm Order button will not appear on the screen.

**Modifying the other transitions in the workflow**

While we have been successful in putting in the art_approval activity and wiring it to router so that the sales order can be processed, we must also modify other transitions. We can begin by deleting the transition that goes directly from the draft activity to the router activity.
Simply click on the transition between the two activities, and then click on the dark circle with the small x to delete the transition.

We have another change that must be made considering the sent activity. If you look at the sales order workflow, you will notice that once you have sent a quote, the sent activity allows you to directly confirm the order. We don't want that. We always want to make sure that the art is approved before we can confirm the order.

Let us change the transition from the sent activity so it can also respond to the approve_art button instead of the confirm button. To modify the transition, double-click on the transition (line) between the sent activity and the router activity.

Then, modify the transition so it is properly connected to the art_approval activity and the signal is the approve_art button instead of the order confirm button.
As you can see from this transition, we still have the **Source Activity** field as sent. However, we have now changed the destination to **approve_art** and the **Signal (Button Name)** field to **approve_art**. We have also specified the **Group Required** field as **Sales / Manager**, so they are the only ones that can approve the art and prepare the order to be confirmed.

With these changes, the workflow will not let us confirm a sales order unless we have first approved the art for that order.

The following is the currently modified sales order workflow to this point:

![Workflow Diagram]

This workflow gets us very close. Notice how while we are in the **draft** mode, we can send a quotation, approve the art, or cancel the order. Once we have sent the sales order, we can approve the art or cancel the order. Once we are at the **approve_art** activity, we can confirm the sales order. There is one transition missing to make the workflow complete. Can you see what it is?

What happens when we are sitting in the **activity** state once the art has been approved? We can confirm the order. Unfortunately, without another transition, we have no way to cancel the order. Let's add the final transition to the workflow to cancel the order.
Using the free black handle at the top of the `approve_art` activity node, drag to the `cancel` activity to create the transition that we need to cancel the order.

Once you save this transition, we can now cancel the quotation after the art has been approved. Our work in the workflow designer is now complete.
In Chapter 13, Discovering Custom OpenERP Modules, we will expand on this workflow and create the necessary custom OpenERP module that will be required to specify the art_approved state inside our sales order model as well as adding the Approve Art button to our sales order form.

If you are a bit adventurous and would like to jump ahead to see how this workflow will be implemented, you can refer to the Extending our module to customize the sales order workflow section in Chapter 13, Discovering Custom OpenERP Modules.

**Summary**

In this chapter, we started by learning about high-level OpenERP workflow views and how they can be used to better understand the processes in OpenERP. Next, we looked at the developer view of the workflow designer and how you can visually analyze and modify the document workflows in OpenERP. After learning about the basic sales order workflow, we modified the workflow to create an approval process for the art designs in our real-world project.

In the next chapter, we will explore how we can use the OpenERP report designer to customize reports and export data from OpenERP. We begin by looking at how to customize our company headers and footers that will appear on our standard documents. With OpenERP, we can use dynamic fields to automatically fill in values from our database into our reports. Finally, we will learn how to use OpenOffice to customize documents in OpenERP to better fit the needs of your business.
Modifying Documents and Reports

Regardless of how great the built-in reports are in any ERP system, it is inevitable that most companies will need to do some custom modifications to the standard documents and reports. Of course, OpenERP is no exception. The goal of this chapter is to provide you with a solid introduction to the OpenERP reporting framework and alternative options for printing more complex reports.

In this chapter we will cover:

- Using RML to modify company headers and footers
- Using OpenOffice to modify a majority of the documents in OpenERP
- Alternative reporting systems that you can integrate with OpenERP

Within OpenERP, it is possible to make some changes without modifying the documents themselves. For example, you can change the headers and footers that appear on all your reports throughout the company. We will begin this chapter by working with the developer tools built in to OpenERP.

In OpenERP Version 7, many of the documents the system creates can be further modified with an OpenOffice plugin. This plugin allows you to connect to the OpenERP server, access the current list of reports, and then bring that list of reports into the OpenOffice writer for editing. After you have made your changes, you can then send the report back to the server. This chapter walks through these steps and shows you how to modify existing OpenERP reports.

Like in other areas of OpenERP development, make frequent backups of your databases. Some of the examples we will show make changes to the database that can be difficult to undo.
In addition to the built-in reports inside OpenERP, it is very likely that you will wish to consider a dedicated report writer. OpenERP's backend database, Postgres, is quite flexible and works great with several open source report designers.

## Company headers and footers

When OpenERP is first installed, you are presented with a default company header and footer that will appear on many of the standard reports. Even if you don't plan to make a lot of major changes to the standard reports in OpenERP, it is very likely you will wish to modify the OpenERP headers and footers to be more specific to your company.

To begin editing OpenERP headers and footers, log in as an OpenERP administrator. Navigate to the **Settings** menu on the far right and select **Companies** in the **Companies** submenu. Click on the **Header/Footer** tab and you will see the XML markup that is used to create the header and footers for your company reports. The following is a screenshot of the Header/Footer tab in the company settings:
For those familiar to RML or XML in general, this code will likely look familiar and much less intimidating. If you are new to XML, you can expect to spend some time looking at other resources that will aide you in making changes to the headers and footers in OpenERP.

I will go on record that it is often the case when end users will believe that with in perhaps an hour or two of training, they will be able to create their own reports. This is not just an OpenERP issue; it is a common perception that many end users have. Unfortunately, creating or modifying reports is often not easy and should be considered more of a developer task than an end user task. Be prepared to spend considerable time to acquire the skills required to make significant changes to the documents and reports in OpenERP.

**RML – Report Markup Language**

Like the header and footer, most of the documents in OpenERP are built in RML. RML is a flexible markup language created by ReportLab Europe Ltd. One of the primary features of this language is it can be used to produce PDF files. Adobe's PDF format is widely accepted, and OpenERP utilizes this format as the primary method of providing documents to the OpenERP end user.

RML provides the template that OpenERP requires to produce the required document. By modifying the RML you can customize reports to your liking. For further information on RML, review the documentation at [http://www.reportlab.com/docs/rml2pdf-ds.pdf](http://www.reportlab.com/docs/rml2pdf-ds.pdf).

**Examining the company header and footer**

OpenERP exposes the company header and footer within the company settings because this will be one of the most frequent changes a company will require. There are three header/footer segments that you can modify:

- The header/footer for documents that will typically be seen externally
- The header/footer for documents and reports that will typically be distributed internally
- The header/footer used for landscape documents

The header and footer information is all within the same text box. The header and footer sections are commented within the code so you know where each section begins and ends.
RML for the company header

Let's look at the RML just for the company header. The following is a screenshot of the RML used for the company header:

```xml
<!-- page header -->
<lines>1.3cm 27.7cm 0cm 27.7cm</lines>
<drawRightString x="20cm" y="27.8cm">[[ company.rml_header1 ]]</drawRightString>
<drawString x="1.3cm" y="27.3cm">[[ company.partner_id.name ]]</drawString>
<place x="1.3cm" y="25.3cm" height="1.8cm" width="15.0cm">
  <para style="main_header">[[ display_address(company.partner_id) or " "]"></para>
</place>
<drawString x="1.3cm" y="26.0cm">Phone </drawString>
<drawRightString x="7cm" y="26.0cm">[[ company.partner_id.phone or " "]]</drawRightString>
<drawString x="1.3cm" y="24.8cm">Mail </drawString>
<drawRightString x="7cm" y="23.6cm">[[ company.partner_id.email or " "]]</drawRightString>
<lines>1.3cm 24.5cm 7cm 24.5cm</lines>
```

Take special notice of the `<!-- page header -->` tag at the very top. It identifies that this segment of RML is utilized to produce the header on the reports. In RML, any text that is wrapped between a `<!--` tag at the beginning and a `-->` tag at the end are considered comments. These will help you identify where you need to make the changes you desire.

Making our first simple change

You have to start somewhere. When modifying reports, the best approach is to start simple and test each and every change as you make it. Do not expect to go in and make a dozen changes to the RML header and then click on Save (not until you have a lot of experience). For our change, notice that in the default header, OpenERP labels the e-mail address Mail:

To see the header, we print any sales order and a PDF is created. The following screenshot shows the default Mail: label used in the company header:
While this is not a big deal, it would be more appropriate to have the e-mail address labeled as Email Address: in our company header.

To make this change, we only need to change Mail: to Email Address: in the header section of the RML and save the record. The following screenshot shows the changed label in the company header:

```
<drawString x="1.3cm" y="25.8cm">Phone <drawString>
<drawRightString x="7cm" y="25.8cm">[company.partner_id.phone or ]</drawRightString>
<drawString x="1.3cm" y="24.6cm">Email Address</drawString>
<drawRightString x="7cm" y="24.8cm">[company.partner_id.email or ]</drawRightString>
<lines>1.3cm 24.5cm 7cm 24.5cm</lines>
```

After you have saved the changes, you can produce the print PDF of another sales order to see the change. The following screenshot shows the changed label in the company header:

As you can see, it is possible to quickly make small changes to the company header without a great deal of concern for all the complex parts of the RML you may not understand.

### RML for the company footer

Now, let's take a look at the RML just for the company footer. The following screenshot shows the RML used for the company footer:

```
<!--page bottom-->
<lines>1.2cm 2.65cm 19.8cm 2.65cm</lines>
<place x="1.3cm" y="0cm" height="2.55cm" width="19.0cm">
  <para style="main_footer">Contact: [:user.name] - Page: <pageNumber></para>
</place>
```

The `<!--page bottom -->` comment tag at the top of this section identifies that the following code is what will be used to display the company footer on the report.
Understanding dynamic fields in your reports

When RML is processed, the text that is between brackets, `[[" and "]], is processed dynamically. This means that the report will not show what exactly is in between the brackets, but instead will fill that data from within OpenERP. In our company footer, for example, the text `[[ user.name ]]` tells the reporting engine to replace that text with the user's name. Take a few minutes to look through the RML and identify where there is information contained in these brackets.

Additionally, this dynamic syntax can call custom Python methods as well. For example, `[[ display_address(company.partner_id) ]]` calls a Python method to correctly format the address for the company header.

Creating a custom footer

By default, OpenERP will automatically create a footer for your report based on the fields you provide in the general information of the company record. If you update information on your company, the information will be automatically updated in your company footer. It is possible, however, to easily create your own custom footer.

To view the report footer configuration, click on the General Information tab. At the bottom, you will see the section for customizing the report footer. The following screenshot shows the company's general information with the report footer configuration:
To create a custom footer, click on **Edit** to edit the form and then select the **Custom Footer** checkbox. The following screenshot shows a custom footer:
In the preceding screenshot we have checked the **Custom Footer** checkbox so that we can edit **Report Footer**. Then we have manually modified **Report Footer** to display the company's Facebook link. Once you have checked **Custom Footer**, any changes you make in the general information of the company record will no longer update the footer. If after making changes to the footer you uncheck the **Custom Footer** checkbox, the footer will once again be generated automatically from the fields in the general information section.

## Modifying OpenERP documents using OpenOffice

Unfortunately, the ability to edit the header and footer in the company information does not get you very far. If you spend time with OpenERP, it is inevitable that the time will come when you need to make changes to specific OpenERP documents. For example, a company may need to customize their quotation or sales order to make it more visually attractive to their customers. Perhaps a company would like to change the appearance of their invoice or the picking ticket they use to pull products from inventory. In these cases, one of the methods to modify these documents is through the OpenOffice plugin for OpenERP.

### Installing OpenOffice

Installing OpenOffice is a prerequisite for using OpenOffice Writer to modify reports in OpenERP. If you do not have OpenOffice installed, there are several resources that can help you properly install OpenOffice on your operating system of choice.

Your primary resource is [www.openoffice.org](http://www.openoffice.org).

Additional resources for OpenOffice are provided in the *Appendix, Locating Additional OpenERP Resources*.

### Installing the OpenERP OpenOffice plugin

To begin modifying documents, you must install the OpenOffice Report Designer for each OpenERP instance. Using the method you learned in *Chapter 2, Starting Your First Company*, install the OpenERP Report Designer application. The following screenshot shows how to install the **OpenOffice Report Designer**.
During the installation, you will be presented with the steps you need to take to make the OpenOffice Report Designer functional in OpenOffice. The following screenshot shows the specific instructions on installing the OpenERP Report Designer:

The steps are remarkably well detailed and break down the two main steps you must take to get OpenERP Report Designer to work properly:

1. Install the plugin inside of OpenOffice.
2. Configure the plugin to work with your OpenERP instance.
Troubleshooting the OpenOffice Report Designer installation

You don't have to unzip openerp_report_designer.zip. OpenOffice recognizes ZIP files as one of the file formats for extensions. Typically, installation goes smooth, but I have personally experienced instances in which you have to reinstall OpenOffice or take additional troubleshooting steps to get the Report Designer to work properly.

Right now, I can verify that the plugin is working successfully in Apache OpenOffice 3.4.1. Depending on what you are trying to accomplish in your specific report modifications, your experience with the Report Designer could be quite frustrating. There are many limitations compared to full featured report writers such as Crystal Reports or even Jaspersoft Reports. Still, even with the limitations, there are many documents in OpenERP that can be easily modified with just a few changes through the OpenOffice Report Designer.

Modifying the sales order

Once you have followed the instructions during the module install and configured the Report Designer for your server, you are ready to begin modifying reports. To modify your report, go to the OpenERP Report Designer menu and select Modify Existing Report.

This will bring up a list of all the reports available in your OpenERP instance for modification. The following screenshot shows how to modify an existing OpenERP report:
For our example, we will select the **Sales Order – Quotation / Order** report. Click on the **Open Report** button to open the report for modification.

If you have been successful in installing the module and configuring the server parameters, you will see a screen similar to the following screenshot of the default OpenERP sales order in the OpenOffice Report Designer:

![OpenERP Sales Order Report](image)

This is the default sales order report that was provided with this instance of OpenERP. It is possible that if you are downloading a different build of Version 7 of OpenERP, your document could look slightly different.
Using OpenOffice Writer to modify the report

OpenOffice Writer is an open source word processing application that serves much the same purpose as a program such as Microsoft Word. If you are familiar with the basic functions of a word processing application, you should not have too much trouble modifying the document itself.

You can change the size of text, make it bold, and with a bit of practice get a good deal of control over placement of the information. However, don't expect for everything to be easy the first time. Even if you are experienced with other word processing applications, expect to take a little bit of time to get used to OpenOffice Writer. Great tutorials on OpenOffice are available at http://www.tutorialsforopenoffice.org/.

Making a small change first

Let's begin by making a small change to the sales order that will make it closer to what we would like our customers to see. When you start making changes to your own reports, it is best to start simple. Save after every change. Backup often. When you break a report, it can often be difficult to troubleshoot what has gone wrong.

Let's look at the default sales order so we can identify what we want to change. The following screenshot shows the default OpenERP sales order in PDF format:

![Default OpenERP Sales Order](image-url)
For better or worse, the default OpenERP sales order gives us an opportunity to make a very simple change that will improve the appearance for many customers. For our example, we will be changing N° to Number:

Using the OpenOffice Writer editor, delete N° and replace it with Number: as shown in the following screenshot:

![Screenshot showing OpenERP Report Designer with code snippets and data fields for Quotation and Order Number]

You will notice that we have made this change in two places. This report uses the exact same report for the quotation as it does for the sales order. Since we want to use Number: for both the sales order and quotation, we have made the change in both places.

**Saving our changes back to the server**

After we have made the modifications, we need to send the modified report back to the server. To send the report to the server, select Send to Server from the OpenERP Report Designer menu. This will bring up a small dialog box. The following screenshot shows the dialog box that appears when sending a report to the server:
For our purposes, we want to accept the defaults. There are instances, however, where you may wish to not include the corporate header in the document. This gives you greater flexibility on that report, but it will require more work because you will need to create a specific header.

Click on **Send Report to Server** and then produce another sales order PDF to see the changes. The following screenshot shows the dialog box that appears when sending a report to the server:
Working with dynamic fields in reports

In our previous example, we were simply changing the static text. Now we will work with a dynamic field from our sales order header. For our real world example, we want to make sure that everyone sees clearly on the sales order when there is a rush order. This was the custom field that we added in Chapter 10, Customizing OpenERP for Your Business. We now want to check this field on our sales order and if the Rush Order field has been checked, we want to display RUSH ORDER in big text on our sales order.

Fortunately, it will be easy for us to add this to our sales order with the skills we have already learned in this chapter. Begin by modifying the sales order in OpenERP. Next, we want to add a line of text that will allow us to only display RUSH ORDER when the x_rushorder Boolean field is true.
The line of markup in our document that adds the rush order message is outlined in yellow. At first, this markup syntax can be confusing. It reads like this; if this is not a rush order, then we want to remove the text that is on this line. The `removeParentNode('paraa')` function hides the RUSH ORDER text. Therefore, when our `x_rushorder` Boolean field is set to True, we will see RUSH ORDER on our report.

**Alternative reporting systems**

OpenERP does a pretty good job of providing basic reports but unfortunately, many companies will require more extensive reporting capabilities than OpenERP provides. Furthermore, the standard tools built into OpenERP to create reports are limited. For example, some things that you would assume would be very simple, such as repeating column headers on multiple pages, can be frustrating if not impossible. Fortunately, there are a variety of reporting solutions that work well with OpenERP.

**OpenERP Webkit Report**

One option that adds a lot of flexibility and integrates very tightly with OpenERP is the OpenERP Webkit Report built by Camp2Camp. It is a standard module that can be installed easily just like the other built-in modules in OpenERP.

Once you install the Webkit Report Engine application, you will have two new menu options appear under **Settings** as follows:

- Webkit Logos
- Webkit Headers/Footers

These options allow you to define multiple headers and footers that you can use for your company reports. Also, Webkit provides additional flexibility in creating different headers for different companies. While not a full feature reporting platform that will give you dramatic flexibility not available in OpenERP, the Webkit Report module can often accomplish simple tasks elegantly.
Jaspersoft reports

If you are working in more of an enterprise environment with complex reporting requirements, you should give Jaspersoft a strong consideration. The product is very robust and flexible. There is a report server that allows for exporting into practically any format you could wish for. Also, there is a feature-filled report editor, iReport, that has one of the most comprehensive sets of tools you will find.

In the previous screenshot, you can see the iReport editor that can be used to design custom reports.

One disadvantage of Jaspersoft is that you will typically work with the database backend in Postgres. This can make for complex queries as you join together the normalized tables in OpenERP. There was a Jaspersoft module in Version 6 but at the time of publication, the module has not been certified for OpenERP 7. Another disadvantage is that you will need to create links and other send parameters to the Jasper Report Server to see your report.
Modifying Documents and Reports

There are some advantages however. Because reports are processed in the Jaspersoft Server, the OpenERP worker threads for the OpenERP application will not have a minimal performance hit when you run the report. Because the queries in Jaspersoft bypass the ORM in OpenERP, it is possibly you could realize much faster performance using Jaspersoft for your reporting.

Additional resources for Jaspersoft are provided in the Appendix, Locating Additional OpenERP Resources.

Translating and reporting data with Pentaho or Kettle

Another potential option to utilize with OpenERP is Pentaho or Kettle. Recently, these products have moved more toward commercial and enterprise offerings. Still, there are community editions available for companies that wish to avoid license and support fees. One of the primary advantages of Pentaho is that it excels at data translations and integrations with other systems. Like Jaspersoft, Pentaho has a server component that makes the Pentaho report services available.

Additional report resources are provided in the Appendix, Locating Additional OpenERP Resources.

Summary

In this chapter we started by walking through how to change the headers and footers on company reports through OpenERP. Next, we went on to defining custom footers for reports. Many documents in OpenERP can be modified using OpenOffice. We walked through installing the module and editing the default OpenERP sales order.

In the next chapter we will dive into creating our own custom modules in OpenERP. With custom modules, we can extend OpenERP with additional fields, automate processes, and better integrate OpenERP for an enterprise operation. Even better, OpenERP's design approach means you will not be making any changes to the OpenERP source code. Instead, you will learn how to extend OpenERP with a module you will build from the ground up!
Discovering Custom OpenERP Modules

While OpenERP has a lot of built-in and community modules, it is inevitable that there will be quite a few businesses that will have requirements which will be difficult to achieve with the currently available modules. The OpenERP framework offers developers the capability of extending OpenERP to accomplish business objectives and (hopefully) make OpenERP fit in better with the workflow of the company. It is important, however, before attempting to write custom OpenERP modules that you completely understand the functionality of OpenERP and the various modules that are available in the community.

Through careful configuration, many business objectives can be achieved without writing custom modules. It is important to make sure that before you go down the path of writing custom OpenERP modules, that you make absolutely sure that the business requirements are clear and you have thoroughly explored all the options available inside OpenERP. There are many settings that provide additional functionality to the OpenERP system. You don’t want to spend days, weeks, or even months building an OpenERP module to then find out that much of that functionality was already available.

The goal of this chapter is to introduce you to custom module development in OpenERP. Even if you are a beginner developer and don’t know much about programming, you should be able to follow along and build a module in OpenERP. If you don’t know Python or XML, you will likely find some aspects of this chapter a little more challenging. Fortunately, there are many resources in the OpenERP community that can help you along your path to OpenERP development.
In this chapter we will cover the following topics:

- Learning the basic structure of an OpenERP module
- Using a module to add additional fields to your OpenERP system
- Extending the views in your OpenERP instance to include new fields
- Making changes to the available states to use in an OpenERP workflow

Exploring the OpenERP application and module directory

In addition to the built-in OpenERP modules and the various settings that can change the way OpenERP functions, there is also a growing collection of custom OpenERP modules written by the community. When you find a business requirement for which you believe you may need to do some custom module development, take the time to go to the OpenERP application repository and search for modules that could perhaps fit the purpose. Even if the module is not exactly what you are looking for, there can often be a lot of valuable code in those modules that can help you with your own module development.

You can find the OpenERP application and module repository at apps.openerp.com.

Building our first OpenERP module

One of the best features of the OpenERP framework is that we can extend OpenERP and write our own modules without having to modify any of the OpenERP source code. Instead, the changes we make are all contained in their own directory and within their own files.

The primary advantage to this point is that when OpenERP modifies their source code with patches or bug fixes, we do not have to worry about our changes getting overwritten. Also, while we may still need to modify our code if OpenERP makes a dramatic change to their source code, there is a reasonable chance the changes required will be minimal.

Like in other areas of OpenERP development, make frequent backups of your databases. Some of the examples we will show make changes to the database that can be difficult to undo.
Each module in OpenERP has basic requirements for it to be properly recognized by the OpenERP framework and installed. Once we successfully install our module, the framework will then extend OpenERP with the appropriate functionality.

Specifying a custom directory to hold our OpenERP modules

We will begin by creating a directory to hold our OpenERP module. We have two options as to where we can create the directory to hold our module. With our first option, we could create our directory in the addons folder where all the rest of the add-ons for OpenERP are stored. This method is easy and allows OpenERP to see our module simply by restarting the OpenERP server.

Alternatively, we could create a separate folder to hold our add-ons. This method has the advantage that we keep our modules separate from the standard OpenERP modules. However, to use this method, we must change the OpenERP configuration to see the alternative directory.

To specify an alternative directory to store our modules, include the following line in the OpenERP configuration file:

    addons_path = /home/openerp/server/addons,/home/mymodules

With the addons_path specified in the preceding line, we could then have a directory named mymodules within our home directory to store our custom OpenERP modules.

After you have decided on the approach you wish to take, create a directory to contain your module. For our example, we will create a directory named silkworm and include it in the standard OpenERP addons directory.

Contents of our module directory

Within our module directory, silkworm, we will create two files that are required in every OpenERP module. These two files must always be named the same:

- __init__.py
- __openerp__.py

Although it is difficult to tell, in both cases, there are two underscores together at the beginning and then another two underscores just before the file extension. You must name these files exactly this way to have a valid OpenERP module.

These are Python files and can be edited with any text editor. We will begin by defining these two required files.
Creating and editing the files

Depending on your operating system of choice, there are a variety of editors you could use to create and edit the files for your module. In Windows, you could use something as simple as Notepad. In Ubuntu there are also several choices, including Nano, Vi, or Vim. Please refer to Appendix, Locating Additional OpenERP Resources, for a list of common editors.

The __init__.py file

The purpose of the __init__.py file is to specify the Python files you wish to include in your module. At minimum, you will usually have one Python file, but you could have more or less depending on the complexity of the module you are developing. If you were to have a Python file with your code and that file was named codexample.py, you would have to import codexample inside the __init__.py file. You will notice you don't have to include the .py extension inside the __init__.py file.

For our example, the __init__.py file will contain one line that specifies the name of the file in which we will be placing the Python code for our module.

```python
import silkworm
```

The __openerp__.py file

The __openerp__.py file is essentially a manifest for your OpenERP module. It describes to the OpenERP framework the necessary attributes of your module. Sometimes, this file is also called the module descriptor file. The structure in the file is what is called a dictionary in Python.

```python
{
    'name': 'Screen Printing',
    'version': '1.0',
    'description': "This module adds functionality for screen printing companies",
    'author': 'Greg Moss',
    'depends': ['base','sale'],
    'data': ['silkworm_view.xml'],
    'demo': [],
    'installable': True,
    'auto_install': False,
}
```
This is how the __openerp__.py file appears when edited in Nano.

The __openerp__.py file contains a single Python dictionary. Even if you don't know Python, the syntax is rather simple if you have had even a little experience in programming. When you install a module in OpenERP, this file describes the details the framework needs to properly configure your module.

**name**

The name entry is what will appear in the modules listing inside OpenERP.

**version**

This allows you to specify a version number for your module. This is valuable as you extend the functionality of your module and need to keep track of the various releases.

**description**

This description will appear when you prepare to install the module in OpenERP. It should clearly describe the purpose of the module to someone who may be entirely unfamiliar with it. You should take time to fill out this entry. Even this little bit of documentation can help someone who is trying to utilize the module in the future.

```
In this example, notice the triple double quotes before and after the description value. Python uses this syntax to allow you to continue a string on multiple lines.
```

**author**

Providing the name of the author of your module is also important, as it could help future users track down the main person who can provide assistance.

**depends**

The preceding elements were pretty self-explanatory and are mostly for documentation purposes. This entry, however, tells the framework what other modules your module will build upon. At minimum, you will need to include base as one of your module dependences. In our example, we will be extending the sales order system, so we have also included sale as one of the module dependencies.
Discovering Custom OpenERP Modules

data
The data item specifies the XML view files you wish to include in your module. We will cover view files in depth later in the chapter. If you plan to change something in OpenERP's forms or user interface, it will most likely involve creating a view file. Other types of data files can be specified here, such as files containing initial data or access rights, but for our example, we have named only the file silkworm_view.xml.

demo
OpenERP provides a rather convenient method of including demonstration data with your module. When you create your database, you have the option to include demonstration data with that OpenERP instance. We have left this blank for our example, but if we wished to make demonstration data available when the module is installed, we could fill in this entry.

installable
This is an entry that you may use to temporarily disable a module for installation. Most often, it will be True because you want the ability to install the module in an OpenERP instance.

auto_install
When this entry is set to True, OpenERP will then automatically install this module when it finds that all the dependency modules are installed. If you have no dependencies, this means that it will be automatically installed when you create a new database. Given OpenERP's modular application approach, you typically would not want to have the auto_install flag set to true for most module development.

Extending an OpenERP model in silkworm.py
Next, we create another file named silkworm.py. We will begin by creating a module that performs the same customizations we performed through the developer mode in Chapter 10, Customizing OpenERP for Your Business.

Why would we want to put our customization into a module rather than just using the developer mode?

First off, changes made through the developer mode are isolated within that instance of OpenERP. If you decide you wish to create a new database, you will have to make all the developer changes manually, again. More importantly, when you make the changes in a module, you have much more control over the final results.
The developer mode is very powerful for quickly looking at views, analyzing fields on forms, and understanding more about the OpenERP framework. However, it is typically far better to make any actual changes by creating a module rather than modifying the views or models in the developer mode.

**Using a module to add custom fields to a model**

In Chapter 10, *Customizing OpenERP for Your Business*, we added **Date Required** and **Rush Order** to our sales order model. Now let’s see how we can do exactly the same thing in our module.

In our `__init__.py` file we only had one line, the import `silkworm` command.

To add the **Date Required** and **Rush Order** fields to our sales order, we can place the following code in the `silkworm.py` file:

```python
from osv import osv, fields

class silkworm_sale_order(osv.Model):
    _inherit = 'sale.order'

    _columns = {
        'x_daterequired': fields.date('Date Required'),
        'x_rush': fields.boolean('Rush Order'),
    }
```

In Python, the `from` command allows you to specify which libraries you wish to utilize in your custom classes. For our simple example, we are only pulling in `osv` and `fields`.

**Inheriting from the sales order module in OpenERP**

In our `class` statement, we specify the class `silkworm_sale_order` and it has the parameter `osv.Model`. Remember that while learning the OpenERP framework, it will take a bit of time to get familiar with the syntax. As of right now, you don’t have to necessarily understand why you are specifying `osv.Model`; just understand that it is required with most classes.

```python
    _inherit = 'sale.order'
```

For those new to object-oriented programming in general, the `_inherit` statement essentially makes available to your class the functionality of the OpenERP sales order module.
Next, we can extend the OpenERP sales order module with our two custom fields:

```python
_columns = {
    'x_daterequired': fields.date('Date Required', required=True),
    'x_rush': fields.boolean('Rush Order'),
}
```

This once again is a Python dictionary. We specify the names of the fields on the left followed by a colon. You will notice in the syntax that we also specify the data types and provide the labels we want to display in the views inside OpenERP. Notice that we have also set `required=True` for the `x_daterequired` field so that the user will be forced to provide this data when they create a sales order record.

### Python conventions

Unlike many programming languages, Python takes white spaces very seriously. In fact, you must exactly indent your code or the Python compiler will generate an error. For example, the `_inherit` attribute and the `_columns` dictionary are indented exactly four spaces over from the `class` command.

### Adding the fields to our sales order view

Now that we have specified the fields we want added to our sales order model, we must now create our view file that will display the fields in the sales order header. We have specified the name of this file inside `__openerp__.py` within the data entry. For our example, the file name is `silkworm_view.xml`.

Using your editor of choice, create the `silkworm_view.xml` file. In this file, place the following code:

```xml
<?xml version="1.0" encoding="utf-8"?>
<openerp>
    <data>
        <record id="sale_view_order_form" model="ir.ui.view">
            <field name="model">sale.order</field>
            <field name="inherit_id" ref="sale.view_order_form"/>
            <field name="arch" type="xml">
                <field name="client_order_ref" position="after">
                    <field name="x_daterequired"/>
                    <field name="x_rush"/>
                </field>
            </field>
        </record>
    </data>
</openerp>
```
OpenERP specifies views using the XML syntax. The first line in the file is the standard element you will find at the top of many XML files, specifying the version and type of encoding used.

Next, OpenERP view files contain beginning and ending openerp tags. Inside those tags, there are matching opening and closing data tags. To modify or add views in your custom OpenERP module, you add the record tags.

Each record must have an id. In this case, we also have a model tag that is specified as ir.ui.view.

```
<record id="sale_view_order_form" model="ir.ui.view">
```

This is a framework convention, and you will learn about other models that are available as you continue to study OpenERP development.

Next, we must specify the base model with which this view interacts. For our example, this is sale.order. This relates directly to the fact that we have added the fields to the sale.order model in our Python file.

```
<field name="model">sale.order</field>
```

If instead you were adding additional fields to the purchase order header, you would specify purchase.order.

Use the developer mode in OpenERP to mouse over fields and determine to which models they relate. To find the view names you need to use, go to Manage Views in the developer mode. This can save you a great deal of time while developing in OpenERP.

Next, let’s look at the line that contains inherit_id.

```
<field name="inherit_id" ref="sale.view_order_form"/>
```

Much like we had to inherit from sale.model when we created our silkworm_sale_order class, we must inherit from the sale.view_order_form view so that we can add the additional fields. One big trick in finding the value you require for ref is to use Manage Views while in the developer mode.
For this example, while on a sales order in OpenERP, choose Manage Views from the developer menu on the left. You will then be taken to the form that shows you exactly the external ID you need to add fields to the form.

The Manage Views screen shown here is available in the developer mode menu.

Essentially, you place the value from External ID into the ref attribute for the _inherit entry.

```
<field name="inherit_id" ref="sale.view_order_form"/>
```

The next entry in the list is required for all view changes.

```
<field name="arch" type="xml">
```

Like other aspects of the framework, you don't need to know exactly why this syntax is used. Just use it.
Now we are ready to get to the segments that specify exactly where we wish to place our new custom fields on the form.

    <field name="client_order_ref" position="after"/>

When you are adding fields to a form, it is important for OpenERP to have the information it requires to determine exactly where the fields should go. In this example, we are telling the OpenERP framework we want to first find the field named `client_order_ref`. Next, we use `position="after"` to specify that we wish the fields to appear after `client_order_ref`.

Once again, we can use the developer mode to visually find the field name we require. Here, we have moved the mouse over the **Customer Reference** field to reveal the details of that field.

Now that we know where to add our fields, we can specify custom fields to display.

    <field name="x_daterequired"/>
    <field name="x_rush"/>
Getting ready to install our module

Right now, our module is very simple and just adds two fields to our sales order form. Still, we should quickly review the files you should have in your module directory:

- __init__.py
- __openerp__.py
- silkworm.py
- silkworm_view.py

To install the module you must restart your server. If you don’t restart your server, OpenERP will not see your module. If you have put the module folder in the OpenERP addons directory where the rest of the modules reside and all of your syntax is correct in the files, there is a good chance you are ready to install the module.

In the top menu, click on Settings and choose Update Modules List from the menus on the left.

On the Module Update screen, click on the Update button. Once you have clicked on Update, OpenERP will refresh the available list of add-ons.

Next, we will install the module. Click on Installed Modules from the menu on the left.

When you create a custom module, do not use Apps under the Modules menu section. This will only show you modules and applications that are available in the OpenERP app repository or are part of the standard OpenERP installation.

Remove the Installed filter from the search box. Yes, this entire process is somewhat counter-intuitive. Just think of the Installed Modules section as the place that truly holds all the available modules including your custom modules.
Once you take off the **Installed** filter, you can search for silk to locate your module for installation.

Click on the **Install** button to begin the installation process.

After a few seconds, the screen will refresh. You can now pull up a sales order and see the fields added to your form.
This screenshot of a sales order shows the custom fields added in our module.

While developing, it is inevitable that a module may not install correctly, or after installing you will have an error that will prevent you from logging in to OpenERP. If you find yourself unable to resolve the error, one workaround to get OpenERP back up and running again is to rename the module directory. This prevents OpenERP from locating the module to install.

**Extending our module to customize the sales order workflow**

In *Chapter 12, Understanding Workflows*, we created a workflow that would require art to be approved before a sales order could be confirmed. Now we will extend our custom module to integrate with that workflow and implement the required business process. While modifying workflows, you are frequently working with the state of the document. As a business process is executed, the system manages the flow of that process by setting the state of the documents involved.

The standard OpenERP workflow allows you three primary options when you create a draft sales order: confirm the draft sales order, send the draft order to a customer by e-mail, or cancel the draft sales order. Depending on your business requirements, there are quite a few other steps involved than what the default workflow provides. For our business example, we wish to have an **Approve Art** stage that must be completed before a sales order can be confirmed.

**Adding a button to the sales order**

In order to approve the artwork, we will need to add a button to the sales order form. This button is then the trigger to set the state of the document to **art-approved**. While it is possible to add a button to your form by going through the developer mode, it is far better to put your code into a module.

Let’s look at the code for *silkworm_view.xml* after we have included the code required to add this button to the sales order.

```xml
<openerp>
<data>
  <record id="sale_view_order_form" model="ir.ui.view">
    <field name="model">sale.order</field>
    <field name="inherit_id" ref="sale.view_order_form"/>

```
Just like when we added the Date Required and Rush Order fields to the form, we can add a button to the form in a similar manner.

```xml
<button name="print_quotation" position="after">
    <button name="approve_art" string="Approve Art" states="draft,sent" groups="base.group_user"/>
</button>
</field>
</record>
</data>
</openerp>
```

This tells OpenERP that we wish to position our Approve Art button after the Print_Quotation button. Between the opening and closing button tags, we place the code for our new button.

```xml
<button name="approve_art" string="Approve Art" states="draft,sent" groups="base.group_user"/>
```

We have named our button approve_art. In our workflow, we specified approve_art as the trigger. Let's look again at how that trigger is specified in the workflow.
Discovering Custom OpenERP Modules

This screenshot depicts the workflow transition from **draft** to **approve_art**.

By adding the button in our form view, we now have the ability to transition from the **draft** node to the **approve_art** node in our workflow. The **Signal (Button Name)** value matches the name attribute of the button in our module.

Notice also that **Source Activity** of the transition is **draft**. In the button code, we have included the attribute `states="draft,sent"`. OpenERP buttons include a `states` attribute that lets you specify in what states the button will be active.

You will want to always make sure that you set the `states` attribute of your button to match the corresponding **Source Activity** field value in your workflow.

For our example, we have specified **draft** and **sent** because those are the states in which we wish the **Approve Art** button to be active.

**Modifying the available states of the sales order model**

Next, we need to update our sales order model to add the **art_approved** state as one of the potential states of the sales order. When we override the list of available states in the sales order, we are going to be required to specify all of the potential states.

While there are better, more sophisticated designs you would use for a production system; for this example, we are going to get the available states from the OpenERP sales order module and then add the state we need to this list. While building your module in OpenERP, it is vital to become familiar with the base modules you are extending. Since we are extending the sales order module, let’s open up the base sales order code and find the `state` column in that class.

**Finding your way around the base OpenERP modules**

OpenERP modules and applications are all stored in the same folder, `addons`. Each module has its own folder and are generally well named and easy to associate with the functionality in OpenERP.
Because we are working in the sales order module, we need to look in the `sale` directory for the code file. OpenERP is also pretty consistent about naming the primary module code the same as the directory, only with a Python extension (`.py`).

After we open up our file in the text editor, we wish to find the `sale_order` class. This is the same class from which we inherited in our module. Next after finding that class, we want to find the column that determines the values for the state. After you have explored a few of the OpenERP classes, it will be easier for you to find where the columns are defined.
Here is the code in `sale.py` that interests us:

```python
'discount': fields.float('Discount (%)', digits_compute= dp.get_precision('Discount'), readonly=True, states={'draft': [('readonly', False)]}),
'th_weight': fields.float('Weight', readonly=True, states={'draft': [('readonly', False)]}),

'state': fields.selection([('cancel', 'Cancelled'), ('draft', 'Draft'), ('confirmed', 'Confirmed'), ('exception', 'Exception'), ('done', 'Done')], 'Status', required=True, readonly=True,
    help='* The "Draft" status is set when the related sales order in draft status. 
* The "Confirmed" status is set when the related sales order is confirmed. 
* The "Exception" status is set when the related sales order is set as exception. 
* The "Done" status is set when the sales order line has been picked. 
* The "Cancelled" status is set when a user cancel the sales order related.'),

'order_partner_id': fields.related('order_id', 'partner_id', type='many2one', relation='res.partner', store=True, string='Customer'),
'salesman_id': fields.related('order_id', 'user_id', type='many2one', relation='res.users', store=True, string='Salesperson'),
```

The `state` column in our model is what we need to extend with our `art_approve` state option. Here you can see the other states that are already available in the sales order:

- cancel
- draft
- confirmed
- exception
- done

We will then copy the `state` column from `sale.py` and paste it into our module. Next, we extend the available states with `art_approve`. 
Discovering Custom OpenERP Modules

Updating the other buttons for the workflow

If you are ambitious and you went ahead and rebuilt your module and ran it, you would quickly find that more work needs to be done. Even though we have added our button to the form and can now actually accept that button into our workflow, we must adjust the states attribute of the other buttons so they are active at the appropriate times.

For example, we do not want the Confirm Sale button available on the screen until the art is approved.

Modifying the attributes of the confirm button

By default, the Confirm Sale button is available when the sales order is in the draft state. Adding the following code to our XML view tells OpenERP to modify the attribute of the Confirm Sale button so it is only available after the art has been approved:

```xml
<button name="action_button_confirm" position="attributes">
    <attribute name="states">art_approved</attribute>
</button>
```

In this example, we have used the position="attributes" attribute (yes, that is a bit confusing) to modify the states attribute of the action_button_confirm button in the default sales order workflow. No longer will the Confirm Sale button be available while the sales order is in the draft state. The sales order must be in the art_approved state for the Confirm Sale button to be visible on the form.

Modifying the attributes of the print and cancel buttons

Modifying workflows will often require you to carefully consider when you want buttons to appear and when you do not want them to appear. For our example, while we are waiting to approve the art, we still want to be able to print the order and cancel the order. With the default Sales Order module buttons, these buttons will not be visible when the state of the sales order is art_approved. Therefore, we must update the attributes of these buttons so they will be visible once the art has been approved.

```xml
<button name="print_quotation" position="attributes">
    <attribute name="states">draft,sent,art_approved</attribute>
</button>

<button name="cancel" position="attributes">
    <attribute name="states">draft,sent,art_approved</attribute>
</button>
```
For each button, we have modified the states attributes to include the art_approved date. It is also important that we include all the other states we wish to have these buttons visible for as well. We are completely replacing the default states with the states from our module.

Modifying workflows and states can have unpredictable consequences on existing documents. Make sure to back up your database. Test your modified workflows carefully and ensure that data integrity while modifying workflows in live production systems.

Seeing it all come together

Now after you save, restart the server, and run your module, the workflow for the sales order will be modified so you can only confirm orders that have had their art approved.
You will notice that not only do the buttons now properly behave with our workflow, but the Art Approved state is also now displayed in the upper-right corner. The user is getting clear feedback that the art has been approved and the sales order is now ready to be confirmed.

**Using a module to add a filter to a search view**

One very nice feature of OpenERP is the flexible but easy-to-use search functionality that is provided on every list view. With a module, you can add additional filter options that make it easier for users to find the information they are looking for. In our real-world example, we have placed an importance on rush orders. Therefore, it would be desirable to have a filter option on our sales order view that will limit our listing to only display rush orders.

Here is how the final search filter view will appear after we implement the module changes:
When **Rush Only** is checked, the sales order list view will limit the orders to only those orders that are specified as rush orders. Quickly, users can now locate rush orders without creating custom filters. This is an example of how a small change can have significant real-world benefits to usability. Best of all, with OpenERP you are making these changes without modifying any of the base OpenERP source code.

### Adding the code to create the rush order filter

The following code segment will naturally be added to the *silkworm_view.xml* file. It will have the same record structure as our other modification. Adding this code segment and updating the module will implement the change we desire.

```
<record id="sale_view_sales_order_filter" model="ir.ui.view">
    <field name="name">sale.order.search</field>
    <field name="model">sale.order</field>
    <field name="inherit_id" ref="sale.view_sales_order_filter"/>
    <field name="arch" type="xml">
        <field name="name" position="after">
            <filter name="rush" string="Rush Only" domain="[('x_rush', '=', True)]"/>
        </field>
    </field>
</record>
```

Let's look at some of the more important elements of this code segment. It follows a similar structure as the modification that added fields to our form. Most important while looking at the code in any modules, is to identify the `inherit_id` field's `ref` value. This is what ties your view modifications to the view in the base module. In this case, our `inherit_id` is `sale.view_sales_order_filter`.

Use the developer mode to lookup the view name from inside OpenERP. Navigate to the view you want to work with, and in the developer menu, you can choose Manage View to see the external ID of the view. You can also use the developer mode to quickly look at the syntax of views and use them to help you determine how your filters should be structured.
Creating the filter

The filter is specified by one line of XML code as follows:

```xml
<filter name="rush" string="Rush Only" domain="[('x_rush', '=', True)]"/>
```

In this code, we specify the name of our filter and the string we wish to display in the search view. The filter is applied with the `domain` parameter. We specify the field from our sales order model and that it must equal to `True` in order for this filter to be valid.

The technical name for this syntax in Python is a **Tuple**. It is possible to include multiple filters in the domain. For example, we can also specify that we only want sales orders that are confirmed by specifying an additional condition in our filter.

```xml
<filter name="rush" string="Rush Only" domain="[('x_rush', '=', True), ('state', '=', 'progress')]"/>
```

OpenERP considers a confirmed sales order to be in the state specified as `progress`. With this change, our rush only filter will also limit the sales orders to only those that are confirmed.

Summary

In this chapter we learned about the basic OpenERP structure for modules. Files must be named exactly the way the OpenERP framework expects, and you must follow the structure for your module to properly load into OpenERP. We explored how to extend OpenERP with additional fields and display them on forms. Next, we extended our custom module with a workflow example. This allowed us to see how we can peek into the existing OpenERP modules to assist us with developing our own module.
Locating Additional OpenERP Resources

OpenERP is built using a variety of open source technologies and components. This appendix includes a list of resources that can extend your knowledge in supporting an OpenERP installation.

Locating the essential OpenERP documentation
If you are looking for the official OpenERP documentation, this is the link for you: https://doc.odoo.com/.

This documentation currently allows you to browse by the OpenERP version and provides direct access to both functional and technical documentation.

Visiting the official OpenERP help site
When you run into issues with your OpenERP installation or you have questions about specific features, one of the best resources available is the official OpenERP help site. The site is simple in such a way that you can search for specific questions, and if you find that your specific question has not been addressed, you can then ask your own question. This is the URL: http://help.odoo.com.
Finding OpenERP applications and modules

OpenERP 7.0 is typically downloaded in a format that already comes with many modules available for installation. However, there are many more modules in the online repository. If you are thinking about making any customization in OpenERP, check out this link to make sure someone else has not already done the work for you: https://apps.openerp.com/.

Getting the latest OpenERP 7 release notes

While it is kind of buried and has a funny URL, this is a good resource for changes in OpenERP 7: http://v6.openerp.com/node/1272.

Downloading OpenERP from Launchpad

OpenERP branches are maintained on Launchpad in a set of projects. The primary OpenERP project is located at https://launchpad.net/openobject.

This project is then broken down into three major project branches:

- https://launchpad.net/openobject-server
- https://launchpad.net/openobject-addons
- https://launchpad.net/openerp-web

Locating resources on Ubuntu

While you can certainly work with OpenERP under the Windows environment, most of the community is in agreement that it is better to run your OpenERP server under Ubuntu. Here are some resources that can help you get started with Ubuntu:

- The official Ubuntu website is http://www.ubuntu.com/
- The direct download URL for the Ubuntu server or desktop is http://www.ubuntu.com/download
- The official Ubuntu documentation is available at https://help.ubuntu.com/
Getting access to the additional developer documentation

The OpenERP framework not only provides a lot of power but also comes with a fairly steep learning curve. While Chapter 13, Discovering Custom OpenERP Modules will help get you started, if you are serious about developing for OpenERP, you will find the resources listed in this section valuable in your pursuit.

Learning from the latest technical memento

The best thing about this resource is it condenses down the most important details into a single, easily downloadable PDF document. While you won't get a lot of examples or explanations, what you will get is a very good overview of the OpenERP framework, which is also useful for a quick reference. This PDF document is available at https://www.openerp.com/files/memento/OpenERP_Technical_Memento_latest.pdf.

Accessing the OpenERP technical documentation

OpenERP keeps a somewhat updated technical developer resource available online. It is also accessible from the main OpenERP website, but you have to dig a bit to get directly to the OpenERP server documentation. This is a must-read resource for expanding your OpenERP development skills. This documentation is available at https://doc.openerp.com/trunk/server/.

Getting quick access to OpenERP installations using OpenERP Runbot

Wouldn't it be great if you could try out a variety of OpenERP builds and installations quickly and easily without going through all the time and complexity of an OpenERP installation? Well fortunately, OpenERP Runbot provides you with many different OpenERP branches that you can connect to and use to actually review the build for yourself.

This resource divides the installations by version, build, and age of the version. Even better, you can get immediate access to the branches right on Launchpad. OpenERP Runbot is available at http://runbot.openerp.com/.
Locating Additional OpenERP Resources

Finding Postgres resources
While supporting any ERP system, it is important to understand the underlying
database architecture. OpenERP 7.0 requires Postgres 9.1 or later as the backend
database. Here is a list of resources that can help you maintain your Postgres server:

- The official Postgres website is http://www.postgresql.org
- Download Postgres from http://www.postgresql.org/download/
- The Postgres documentation is available at http://www.postgresql.org/docs/9.1

Locating Python resources
OpenERP modules are created in Python, a free and powerful programming
language. Most module development will require you to at least become familiar
with the basics of Python. OpenERP requires Python Version 2.7 or higher. Here
are some valuable resources for Python:

- The official Python website is http://www.python.org/
- Download Python from http://www.python.org/download/
- The Python documentation is available at http://docs.python.org/2/

Finding XML resources
OpenERP views are designed and maintained in XML (Extensible Markup
Language). A basic understanding of XML will help you customize views,
modify search criteria, and manage workflows. Here is a list of XML resources:

- The World Wide Web Consortium XML resource page is
  http://www.w3.org/XML/
- XML tutorials and documentation are available at http://www.w3schools.com/xml/

Locating RML resources
RML (Report Markup Language) is a specialized XML format utilized for
reporting. OpenERP uses RML for most forms and simple reports inside OpenERP.
Understanding RML basics will help you design reports or modify headers and
Alternative reporting solutions
Many companies will want more reporting options that are currently not available in OpenERP. Like many other ERP systems, companies will typically utilize and integrate alternative reporting platforms. Here are a few that are popular in the OpenERP community.

Jaspersoft reports
Jaspersoft offers a variety of both open source and licensed reporting solutions. There is a report server as well as a very nice graphical report editor available.

Community edition of JasperReports Server

Community edition of iReport Designer
The server software provides report access and processing. To design reports graphically, you will need to download the iReport Designer for the operating system of your choice from http://community.jaspersoft.com/project/ireport-designer.

Pentaho/Kettle
Pentaho, also known as Kettle, is a very useful tool for data translations and for reporting out to Excel or other formats. It doesn't provide the extensive reporting tools of Jasper, but is very robust for integrating OpenERP with other systems and creating automated data exports. Depending on your needs, Pentaho may fit your needs. You can get this from http://kettle.pentaho.com/.

Aeroo Reports
Another popular reporting option that integrates well with OpenERP is Aeroo Reports for OpenERP. This solution doesn't have the robust reporting ability of Jasper, but integrates better with the OpenERP framework. It is an option that you may wish to consider if the built-in OpenERP reports do not meet all your needs. You can find a lot more in detail on Aeroo Reports at https://launchpad.net/aeroo.
Symbols

&debug=  216
.dump file  219
__init__.py file  276
__openerp__.py file
about  276
author  277
auto_install  278
data  278
demo  278
deps  277
description  277
installable  278
name  277
version  277
& symbol  218

A

access
  setting, to Menus tab  178, 179
  setting, to Views tab  178, 179
access rights
  determining  180, 181
Access Rights tab  83
account
  specifying, for product category  150, 151
  specifying, for sales tax  148
Accounting and Finance application
  installing  138
accounting data
  configuring  50
accounting options
  setting  50

Accounting page
  about  57, 65
  Account Payable field  59
  Account Receivable field  58
  Bank Accounts option  60
  Credit Limit field  59
  Customer Payment Term option  58
  Fiscal Position field  58
  Latest Reconciliation Date field  59
  Supplier Payment Term field  59
  Total Receivable field  59
accounting transactions
  tracking, through sales and accounts receivable process  142, 143
account manager
  assigning  80, 81
  time zone, designating  81
Account Payable
  transactions, creating  140
accounts
  setting up  149, 150
accounts setup
  about  149
  account, specifying for product category  150, 151
  fiscal year, configuring  152
  journal entries, creating  153, 154
  period, closing  153
  period, configuring  152
Active field  89
activity node
  adding, to workflow  245, 246
advanced search
  performing  207, 208
  saving  209, 210
Aeroo Reports 301
all-in-one OpenERP installation
performing, in Windows 35-38
alternative reporting solutions
about 301
Aeroo Reports 301
Jaspersoft reports 301
Pentaho/Kettle 301
alternative reporting systems
about 270
Jaspersoft reports 271, 272
OpenERP WebKit Report 270
Pentaho 272
art approval
restricting, to sales manager 248, 249
Attach a File feature 92
automated clearing house (ACH) 60
Basics, project management 183, 184
bill of materials
about 125
building 125, 126
production, confirming 126
browser destination
checking 41, 42
browser requisites, OpenERP 18
built-in access rights
knowing 175-177
business document
call, scheduling 93
following 92, 93
button
adding, to sales order form 286-288
updating, for workflow 292
Cancel button
attributes, modifying 292
Chart of accounts
defining, for business 137
Cloud 17
Community edition
of iReport Designer 301
of JasperReports Server 301
Company footer
examining 257
modifying, RML used 256, 257
Company goals
used, for CRM system design drive 79
Company headers
examining 257
modifying, RML used 256, 257
Company information
saving 67
setting 65, 67
configuration, accounting data
about 50
accounting options, settling 50
Configuration page 67
Confirm button
attributes, modifying 292
Coordinated Universal Time (UTC) 34
cost price
manufacturing lead time, estimating 109
records, setting to active 109
setting, of product 109
Create button 81
Creative Designs team 100
CRM
about 21, 56, 77
using, as business strategy 77, 78
CRM application
installing 80
CRM, as business strategy
about 77
account manager, assigning 80
CRM application installations 80
customer experience, improving 79
customer experience, managing 78
leads, tracking 86
opportunities, tracking 86
personalized customer experience,
creating 78
sales representative, assigning 80
CRM system design
company goals, using 79
mission statements, using 79
Current chart of accounts
viewing 139
Current timesheet 163
customer
  editing 55
  salesperson, assigning to 85
  selecting 68

customer creation
  about 53
  Is a Company checkbox 54

customer experience
  improving 79
  managing 78

customer record
  saving 60

Customer Relationship Management systems. See CRM

custom field
  adding, to Sales Order form 220-225

custom footer
  creating 260-262

custom-printed t-shirt production example
  about 122
  bill of materials, building 125, 126
  manufacturing order, creating with work center 133
  manufacturing resource planning (MRP), installing 123
  product, producing 127
  routing orders, creating 132
  work center, creating 129

dashboards
  about 210
  information, adding to 211, 213
data
  entering, into OpenERP form 54
database
  backing up, in OpenERP 218, 219
  creating, in OpenERP 46, 47
  managing, in OpenERP 49
  restoring, in OpenERP 219, 220
developer mode
  activating 215-233
  getting out 217, 218
dictionary 276
directory
  creating, for OpenERP modules 275
double-entry accounting 141
draft invoice
  creating 117
dynamic fields
  using, in reports 260
  working with 269, 270

E
Edit option 90
e-mail
  address, specifying 82
  configuring, settings 82
  messages, receiving via 82
Email and Phone fields 88
employee
  configuring, for entering timesheets 165
  entering, into OpenERP 158, 159
  hiring 175
Employee Directory application
  installing 158
employee hours
  tracking, with timesheet 166, 167
employee, Personal Information tab 162
employment application
  creating 174
ERP system
  implementing, with modular approach 46
Escalate button 99
example, OpenERP workflow
  about 232
  actual step, viewing in workflow 234
  developer mode, activating 232, 233
  help, obtaining from OpenERP 234
Expected Revenue field 87

F
fields
  adding, to Sales Order list view 228, 229
  adding, to sales order view 280-283
fields, Public Information tab
  Coach 161
  Department 160
  Job 161
  Manager 161
  Related User 160
  Visibility 160
filters
using, in list views 203, 204
Following button 93
Function field 88

H
held 94
help
obtaining, from OpenERP 234
human resource application
modular approach 157

I
information
adding, to dashboard 211, 213
grouping, in lists 205, 206
Information page
EAN13 Barcode option 62
Internal Reference option 62
Product Type option 62
Sale Price option 62
inherited groups 177, 178
installation, Accounting and Finance application 138
installation, Employee Directory application 158
installation, manufacturing resource planning (MRP)
about 123
manufacturing order, creating 123, 124
installation, OpenERP
on Ubuntu 40
on Windows OS 32, 33
installation, OpenOffice 262, 263
installation operating system
selecting 28
installation, project management application 184
installation, purchasing application 105
installation, Sales Management module
about 49, 50
accounting data, configuring 50
Internal Notes field 88, 91
Interview button 173
inventory analysis 127

Inventory page
about 63
Product Manager field 64
Status field 64
invoice
creating 74
settling 104, 105
iReport Designer
community edition 301

J
Jasper 301
JasperReports Server
community edition 301
Jaspersoft 301
Jaspersoft reports 271, 272
journal entries
creating 153, 154
journal items
viewing 141

K
kanban view
opportunity, viewing 90
Kettle 301

L
Launchpad
OpenERP, downloading from 298
leads
example 86
leads, OpenERP
creating 98, 99
Lead tab
about 88
Active field 89
address, entering 88
contact information 88
Opt-Out checkbox, Mailings 89
references, specifying 89
Referred By field 89
leave management
department field 169
duration field 169
employee field  169  
leave type field  169  
mode field  169  
leave management application  
   about  168  
   leave request, creating  168  
   leave requests, approving  170  
   Leaves Summary option  170  
   leave, submitting for approval  169  
leave request  
   approving  170  
   creating  168  
Lil League  184  
line items  
   entering, on sales order  70  
line items, entering on sales order  
   product description  71  
   product quantity  71  
   product, selecting  70  
   taxes  71  
   unit price  71  
lists  
   information, grouping in  205, 206  
list views  
   filters, using  203, 204  
Log a note button  92  
Long Term Support (LTS) versions  31  

Microsoft Windows OpenERP installation  
   selecting  29  
Model-View-Controller design  
   about  27  
   controllers, authoring  28  
   models, designing  28  
   views, rendering  28  
module descriptor file  276  
module directory  
   contents  275  
multiple advanced searches  
   specifying  208  

N  

Nano  40  

O  

object relational database management system.  See ORDMS  
OpenChatter feature, OpenERP  
   opportunity, managing  91, 92  
   using  91  
OpenERP  
   about  17, 105, 297  
   access, obtaining to additional developer documentation  299  
   accounting data, configuring  22  
   accounting options, setting  23, 24  
   application repository, exploring  274  
   application, selecting  21, 22  
   applications, finding  298  
   browser requisites  18  
   customer, creating  52, 53  
   customizing  215  
   database, backing up  218, 219  
   database, creating  46, 47  
   database, managing  49  
   database, restoring  219, 220  
   developer mode, activating  215-217  
   developer mode, getting out  217, 218  
   documentation, locating  297  
   downloading, from Launchpad  298  
   employee, entering into  158, 159  
   free online trial, accessing  18-20  
   help site, visiting  297
installations, troubleshooting 41
installing, on Ubuntu 40
installing, on Windows OS 32, 33
leads 86
leads, creating 98, 99
management tips 41
modules, exploring 274
modules, finding 298
OpenChatter feature, using 91
opportunity 86
product, entering into 60
Sales Order form, adding custom field 220-225
searching feature 200-203
signing up, for continued service 25
technical documentation, accessing 299
upgrading 31
URL, for documentation 297
URL, for help site 297
user language, designating 81
user language, selecting 81
using, without subscription fees 26
OpenERP 7
release notes, obtaining 298
URL, for release notes 298
OpenERP architecture
about 26
code, writing with Python 27
database specifications, observing 26
Model-View-Controller design, following 27
PostgreSQL database, introducing 26
OpenERP configuration file
modifying 43
OpenERP documents
modifying, OpenOffice used 262
OpenERP form
database name, selecting 47, 48
data, entering into 54
default language, specifying 48
demonstration data, loading 48
master password, setting 47
password, selecting 48
OpenERP installation
access, obtaining with OpenERP Runbot 299
testing 40
OpenERP installation, on Ubuntu
sources.list, modifying 40
OpenERP interface
about 51
search box, using 52
views, picking 52
OpenERP model
extending, in silkworm.py file 278
OpenERP module
building 274
code, adding for rush order filter
creation 295, 296
contents 275
directory, creating for 275
extending, to sales order workflow
customization 286
files, creating 276
files, editing 276
installing 284-286
used, for adding custom field to model 279
used, for adding filter to search view 294, 295
OpenERP online
benefits 18
getting started 17
OpenERP releases 31
OpenERP Runbot
access, obtaining to OpenERP installations 299
OpenERP services
starting, in Ubuntu 42
stopping, in Ubuntu 42
OpenERP WebKit Report 270
OpenERP workflow
changes, planning 234
eample 232
tasks 235
OpenOffice
installing 262, 263
URL, for tutorials 266
used, for modifying OpenERP documents 262
OpenOffice Report Designer installation
troubleshooting 264
OpenOffice Writer
about 266
used, for modifying report 266, 267
Opportunities button  86
opportunity
  about  86
  categories  87
  converting, into quotation  98
  creating  86
  customer  87
  Email and Phone fields  88
  example  86
  expected closing  87
  Expected Revenue field  87
  Internal Notes area  88
  next action, taking  87
  percentage  87
  priority  87
  subject  87
opportunity, in kanban view
  action, taking  90
  viewing  90
Opt-Out checkbox  89
order
  delivering  122
ORDMS  26
OS option
  selecting, for OpenERP  31
Other Information tab  98
Overdue Payments page  67

P
Pentaho
  about  301
  used, for translating data  272
personalized customer experience
  creating  78
Postgres
  URL  300
  URL, for documentation  300
  URL, for downloading  300
PostgreSQL
  about  26
  URL  27
Postgres resources
  finding  300
posting transactions
  practicing  147
primary OpenERP log file
  searching  42
print button
  attributes, modifying  292
Probability field  96
procurement
  product, configuring for  107
procurement method
  about  108
  defining  109
Procurements page
  about  63
  Cost option  63
  Description for Suppliers option  63
product
  adding, to quotation  112
  configuring, for procurement  107
  cost price, setting  109
  entering, in OpenERP  60
  producing  122, 127
  receiving  104, 114-116
  saving  65
  selecting  124
  suppliers, assigning to  110
product category
  about  61
  account, specifying for  150, 151
product code
  designating  110
product, configuring for procurement
  about  107
  procurement method, defining  108, 109
  supply method  109
production
  confirming  126
product name
  about  61
  designating  110
product, producing
  inventory, analyzing  127
  routings, managing  128
  work orders, managing  128
project
  creating  185, 186
  info, setting  187
  stages, defining for  188, 190
  team members, adding to  186, 187
project costs
  estimating 196

project management
  basics 183, 184

project management application
  installing 184

project stages
  about 188
  defining, for project 188, 190

project tasks
  about 190
  additional task information, defining 191
  additional tasks, creating 192-195
  completing 196
  defining 191

project time
  estimating 196

Public Information tab
  fields 159-161

purchase order
  confirming 113, 114
  generating 104

purchase quotation
  creating 111, 112

purchasing application
  installing 105

purchasing process
  about 103
  invoice, settling 104, 105
  product, receiving 104
  purchase orders, generating 104
  quotations, generating 104
  supplier, setting up 103
  warehouse locations, setting up 104

Python
  URL 27, 300
  URL, for documentation 300
  URL, for downloading 300

Python conventions 280

Python resources
  locating 300

Q

Quality Assurance 188

quotation
  generating 104

printing 113
  product, adding to 112

R

real-world project example 185

recruitment process application
  about 171
  employees, hiring 175
  employment application, creating 174
  Interview button 173
  recruitment, for new job 172
  recruitment stages, defining 171

Referred By field 89

Related Status field 96

report
  modifying, OpenOffice Writer used 266, 267

Report Markup Language. See RML

report modifications
  saving back to server 267, 268

reports
  dynamic fields, using 260

resources
  locating, on Ubuntu 298

RML
  about 257, 300
  URL, for documentation 257
  used, for modifying company
    footers 256, 257
  used, for modifying company
    headers 256, 257
  using, for company footer 259
  using, for company header 258, 259

RML resources
  locating 300

routing orders
  creating 132

ruleflow. See Drools Flow

S

sale
  invoicing 74
  selecting 97, 98

sale negotiations 94, 95
sales and accounts receivable process
accounting transactions, tracking through 142, 143

sales application settings
Manager 84
User: All Leads 84
User: Own Leads Only 84
Sales Department 83

Sales Management module
installing 49, 50

sales order
entering 67
line items, entering on 70
modifying 264, 265

sales order, entering
Customer Reference field 69
customer, selecting 68
Date field 69
quotation, saving 72

sales order form
another field, adding to 226, 227
button, adding to 286-288
custom field, adding to 220-225

Sales Order list view
fields, adding 228, 229

sales order model
states, modifying 288

sales order state
modifying, in activity 246

sales order workflow
about 72, 237, 238
Cancel Quotation button 73
Confirm Sale button 73
editing 243, 244
Print button 73
Send by Email button 73
visual workflow designer, accessing for 235, 236

Sales page 64

salesperson
assigning, to customer 85

Sales & Purchases page
about 35
Active flag 57
Customer checkbox 56
Date field 56

Language field 56
Opt-Out flag 57
Receive Messages by Email option 57
Reference field 56
Salesperson field 56
Supplier checkbox 56

sales representative
assigning 80, 81

sale stages
modifying 95-97

sales tax
account, specifying for 148

sales tax liability
viewing 147

sales team stages
configuring 100

scheduled calls
checking 94

searching feature, OpenERP
about 200-203
advanced searches, saving 209, 210
advanced search, performing 207, 208
filters, using in list views 203, 204
information, adding to custom dashboard 211, 213
information, grouping in lists 205, 206
multiple advanced searches, specifying 208

search requirements
determining, for business 199, 200

signature section 83

Silkworm 46

silkworm.py file
OpenERP model, extending 278

sources.list modification
package, installing 40

status
promoting, to RFQ (request for quotation) sent 113

supplier
assigning, to product 110
establishing 110
setting up 103, 106, 107

supplier assignment, to product
delivery time, calculating 111
minimal quantity, setting 111
product code, designating 110
supplier, establishing 110
supplier product name, designating 110
**supplier companies, versus individuals**
   designating 107
**supplier invoices**
   paying 117, 118
**supply method** 109

**T**

**team members**
   adding, to project 186, 187

**timesheets**
   account setup, for tracking time 164, 165
   employee hours, tracking with 166, 167
   managing 163
   validating 167

**time zone**
   Access Rights tab 83
   designating 81-85
   e-mail address, specifying 82
   message, receiving via e-mail 82
   sales application settings, selecting 84
   Sales Department 83
   signature section 83
   user password, manually setting 84

**transactions**
   creating, in Account Payable 140
   viewing 144-147

**transitions**
   modifying, in workflow 249-253

**Tuple** 296

**U**

**Ubuntu**
   OpenERP, installing on 40
   resources, locating on 298
   URL 298
   URL, for documentation 298

**Ubuntu OpenERP installation**
   selecting 29, 30

**user access rights**
   working with 175

**user password**
   setting, manually 84

**V**

**Views tab**
   access, setting to 178, 179

**visual workflow designer**
   accessing, for Sales Orders workflow 235, 236

**W**

**warehouse locations**
   setting up 104

**Windows**
   all-in-one OpenERP installation, performing 35-38

**Windows installation**
   testing 39

**Windows OS**
   OpenERP, installing on 32, 33

**work center**
   creating 129
   resource type, defining 130, 131

**workflow**
   activity node, adding to 245, 246
   defining, for business 122
   ending 242
   modifying 243, 244
   step, adding for art of approval 243
   transition, defining 247, 248
   transitions 241
   transitions, modifying 246, 247

**workflow nodes**
   about 237
   examining 239, 240
   sales order workflow 237, 238

**X**

**XML (Extensible Markup Language)** 300

**XML resources**
   finding 300

**x_prefix** 221, 224
Thank you for buying
Working with OpenERP

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 Open Source
In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

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.
Microsoft Dynamics NAV 7 Programming Cookbook

Learn to customize, integrate and administer NAV 7 using practical, hands-on recipes

1. Integrate NAV with external applications, using the C/AL or SQL server
2. Develop .NET code to extend NAV programming possibilities
3. Administer the Microsoft NAV 7 server and database

Oracle PeopleSoft Enterprise Financial Management 9.1 Implementation

An exhaustive resource for PeopleSoft Financials applications practitioners to understand core concepts, configurations, and business processes

1. A single concise book and e-book reference to guide you from PeopleSoft foundation concepts through to crucial configuration activities required for a successful implementation
2. Real-life implementation scenarios to demonstrate practical implementations of PeopleSoft features along with theoretical concepts

Please check www.PacktPub.com for information on our titles
ADempiere 3.6 Cookbook
Over 100 recipes for extending and customizing
ADempiere beyond its standard capabilities

1. Extend and customize ADempiere to meet your business needs
2. Integrate ADempiere with popular shopping carts such as Webstore, VirtueMart, and PayPal
3. Packed with recipes enabling you to learn about ADempiere best practices
4. Effectively integrate ADempiere with business-critical tools such as e-mail clients

Getting Started with Dynamics NAV 2013 Application Development
A simple and practical guide to creating a relevant application for your company using Dynamics NAV 2013

1. Understanding user requirements and drawing inspiration from existing functions
2. Creating the application and integrating it into standard Dynamics NAV
3. Presented in a simple tutorial style, with a resource to get a free trial full version to help you get started

Please check www.PacktPub.com for information on our titles