
OpenERP 应用和开发基础 (0.2 版)

作者：老肖

msn: saiwai76@hotmail.com

mail: xiangfu.xiao@gmail.com

QQ: 1417063315

更新日期：

2010 年 1 月，完成 0.1 版，即第六至第十章内容

2010 年 3 月，完成 0.2 版，即第一至第五章内容

2010 年 4 月，全面修订后全文发布

目录

第一章	OpenERP 介绍	1
第一节	OpenERP 的历史	1
第二节	OpenERP 的架构	2
一、	OpenERP 应用系统架构	2
二、	OpenERP 应用服务器架构	3
三、	PostgreSQL 数据库简介	3
第三节	OpenERP 功能和应用简介	4
一、	OpenERP 功能模块	4
二、	OpenERP 的用户群	7
第四节	OpenERP 开发简介	7
一、	数据访问功能开发	7
二、	工作流和报表开发	9
第五节	为什么选择 OpenERP	12
第二章	进销存业务概述	14
第一节	采购业务概述	14
第二节	销售业务概述	15
第三节	采购、销售的财务处理	16
第四节	仓库管理业务概述	18
第三章	OpenERP 基础业务设置	19
第一节	业务伙伴信息设置	19
一、	理解业务伙伴	19
二、	业务伙伴联系信息	20
三、	业务伙伴购销信息	21
四、	业务伙伴事件	22
五、	业务伙伴财务信息	22
第二节	产品信息设置	23
一、	产品模块配置	23
二、	产品一般信息	26
三、	产品供应和仓储信息	27
四、	产品价格和供应商信息	28
五、	产品描述和包装	29
六、	产品财务信息	30
第三节	财务数据设置	30
一、	会计年度设置	31
二、	会计科目表设置	31
三、	分类账 (Journal) 设置	33
四、	税种设置	34
第四节	库存信息设置	36
一、	理解 OpenERP 的库存管理	36
二、	库位和仓库设置	37
三、	设置最小库存规则	38
四、	库存移动 (stock_move)	39
五、	装箱单 (Packing List)	40
六、	库存盘点	41

第四章	进销存案例实施	42
第一节	模块安装	42
第二节	财务数据设置	44
一、	设置会计科目表	44
二、	设置分类账 (Journal) 和税种	46
第三节	业务伙伴和产品信息设置	47
第四节	采购业务测试	47
一、	采购订单	47
二、	采购财务处理	49
三、	采购付款	51
四、	采购入库处理	52
第五节	销售业务测试	53
一、	销售订单	53
二、	销售订单财务处理	56
三、	销售出库处理	58
第六节	财务报表确认	58
第五章	OpenERP 数据导出和导入	60
第一节	数据库备份和恢复	60
第二节	CSV 数据导出导入	60
一、	数据导出	61
二、	数据导入	62
第三节	会计科目数据导入	64
一、	科目类型导入	64
二、	会计科目导入	65
第四节	XML 数据导入导出	68
一、	XML 数据导入	68
二、	XML 数据导出	69
第五节	PostgreSQL 数据库数据导出导入	69
第六章	OpenERP 开发入门	72
第一节	基本概念	72
第二节	实例演练	74
一、	创建请假单对象	74
二、	创建视图	75
三、	创建菜单和系统动作 (Action)	77
四、	测试	79
五、	视图编辑器	80
第三节	深入数据库	81
第七章	OpenERP 的工作流开发	84
第一节	理解工作流 (Workflow)	84
第二节	实作工作流	85
一、	修改对象和视图	85
二、	创建工作流、活动和迁移	87
三、	测试工作流	91
第三节	复杂工作流实例分析	91
第四节	企业流程 (Enterprise Process)	93
第八章	OpenERP 的报表开发	96

第一节	OpenERP 报表开发的预备知识	96
一、	报表开发工具	96
二、	XSL 简介	97
第二节	OpenERP 报表简介	98
一、	基于 sxw 或 odt 模板的报表	98
二、	基于 xsl+xml 模板的报表	99
第三节	报表中文化解决方法	100
一、	让报表显示汉字	100
二、	让报表说中文	101
第四节	实作报表	101
一、	编写 RML 文件	101
二、	编写报表对象代码及 XML 文件	102
三、	测试	103
第九章	OpenERP 的权限设置	105
第一节	OpenERP 权限机制简介	105
第二节	组的权限设置方法	106
第三节	OpenERP 的预设权限组(Group)	108
第四节	Domain 条件表达式	109
一、	Domain 条件表达式的应用场合	109
二、	Domain 条件表达式写法	111
第十章	OpenERP 翻译方法	112
第一节	资源术语翻译	112
第二节	通用术语翻译	114
第三节	批量翻译或纠正翻译	115
第四节	OpenERP 国际化原理浅析	118
第十一章	OpenERP 模块开发	120
第一节	OpenERP 模块结构	120
一、	安装示例代码	120
二、	请假模块结构	121
第二节	理解 OpenERP 的代码	122
一、	对象定义	122
二、	视图、动作及菜单定义	124
三、	工作流和报表定义	125
四、	权限定义	125
第三节	服务端动作(Server Action)	125
一、	配置 Server Action	125
二、	配置 OpenERP Mail	126
三、	测试通知 Mail	127
四、	更多 Server Action	128
第十二章	超强的继承功能	130
第十三章	基于 XML-RPC 的集成	130

第一章 OpenERP 介绍

OpenERP 是一款基于 Python 语言的开源 ERP 和 CRM 软件。它和 Java 软件一样，是跨平台的，而且，同时支持 C/S 和 B/S 两种架构形式。在官方标准发布版中有 100 多个模块。涵盖财务管理、采购/销售管理、库存管理、MRP、CRM、人力资源管理、销售点管理、项目管理等众多功能。在 OpenERP 的开源社区上，目前有超过 500 个由开源社区贡献的模块，涵盖各个行业的管理解决方案。

本章将先介绍 OpenERP 的历史，其次介绍 OpenERP 的功能概要，而后介绍 OpenERP 的架构和开发概要，最后介绍 OpenERP 和别的 ERP 软件的比较。

第一节 OpenERP 的历史

OpenERP 由比利时的“比尔·盖茨”Fabien Pinckaers 所创建。Fabien Pinckaers 现年 30 岁，他 13 岁开始学习编程，2002 年当他还在 Louvain-la-Neuve 大学土木工程系学习时，创办了 Tiny Sprl 公司。Tiny Sprl 公司总部位于有 150 年历史的比利时城市 Louvain 的近郊，那里有比利时最大的计算机学院，法语鲁文大学（Catholic）计算机系。下图为 Fabien Pinckaers 的近照，旁边的白色房子即为 Tiny SPRL 公司的总部。



Tiny Sprl 公司的第一个产品就是开发 Tiny ERP，即后来的 OpenERP。OpenERP 最初由 Fabien Pinckaers 一个人开发，现在，截至 2009 年底，Tiny Sprl 公司有 75 名专职开发人员。官方发布的标准功能模块超过 100 个，由开源社区 Open Object Community 提供的功能模块超过 500 个。OpenERP 的功能涵盖财务管理、采购/销售管理、库存管理、MRP、CRM、人力资源管理、销售点管理、项目管理等众多方面。OpenERP 已经不仅仅是一个 ERP 软件，而是涵盖各个行业的、非常容易定制化的全面管理解决方案。OpenERP 的核心和所有模块都是开放源代码的，你可以自由使用、修改和发布，只要你也保证开源即可。

现在，截至 2009 年底，OpenERP 的用户群遍及 45 个国家，每天的下载量超过 1000 份。始于 2008 年的金融危机，各公司都开始考虑低成本的管理软件，给 Tiny SPRL 公司极大机遇。2009 年上半年开始，Tiny SPRL 公司的每个月的盈利以 10% 以上的速度增长。2009 年上半年，公司盈利 60 万欧元。2009 年底，公司获得 300 万欧元风投融资，结合自身资金，公司预计 2010 年投入 400 万欧元扩张规模。商业杂志趋势（Trends）预估，Tiny Sprl 公司的获利将在 2011 年攀至 1050 万欧元。

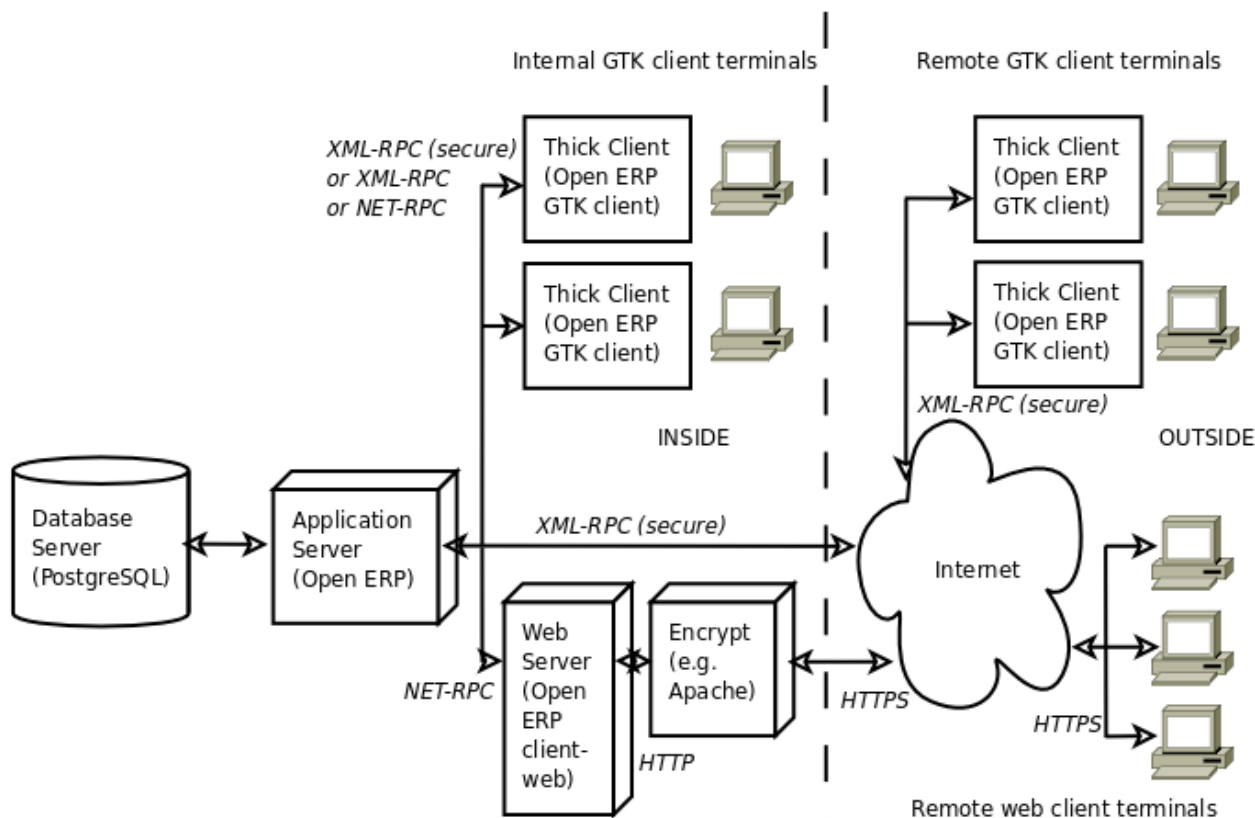
中国 OpenERP 社区，从 2007 年开始介绍 OpenERP 到中国，主要论坛在这里：<http://www.shine-it.net/>。目前中国 OpenERP 社区已经完成了 OpenERP 主要模块的中文翻译工作，完成了报表的汉化，翻译了大部分的用户手册。工作成果参看这里，HornERP 开源项目：<http://code.google.com/p/hornerp/>。

第二节 OpenERP 的架构

一、 OpenERP 应用系统架构

OpenERP 应用系统，大的来说，分成三层，即数据库服务器、应用服务器、客户端。

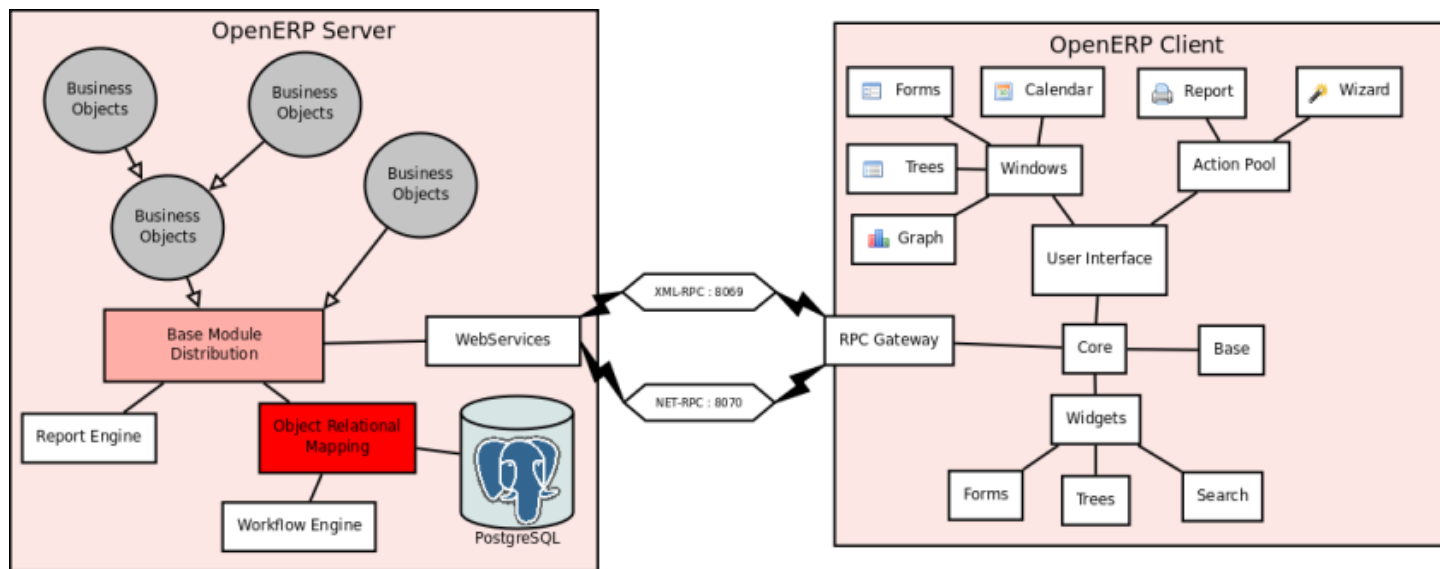
- **数据库服务器**，OpenERP 采用开源数据库服务器，PostgreSQL database server。数据库服务器存放所有业务数据和绝大部分配置数据。OpenERP 的配置数据和一般说的软件配置数据不同，如用户界面显示，每个界面上要显示什么对象的哪个字段，按什么风格显示，在 OpenERP 也是配置数据。在 OpenERP 中，用户界面不是通过 HTML 或其他程序代码生成，而是直接存放在数据库中。运行时直接从数据库中提取数据，渲染成你看到的界面。
- **OpenERP 应用服务器**，应用服务包含所有业务逻辑代码。包括最简单的 CRUD 代码，以及复杂的查询条件等。业务逻辑由各业务对象实现，你可以通过远程调用（Net-RPC 或 XML-RPC）直接调用对象的方法。如大多数业务对象都有 CRUD 四个方法，你可以远程调用对象的 CRUD 方法，从而实现数据库访问。
- **GUI-Client**，这是一个客户端，直接连接在应用服务器上。OpenERP 官方提供的 GUI-Client 是 GTK-Client，即基于 GTK(Graphics Tools Kit)图形界面工具包开发的界面软件。开源社区也有人在开发基于 QT 图形界面工具包的客户端软件。OpenERP 的客户端不含任何业务逻辑代码，也不包含任何界面代码，即所谓的“瘦客户端”。OpenERP 的 Client 的功能是，通过调用 OpenERP 应用服务器的各对象的方法，从数据库中取得界面数据（XML 格式定义的界面），以及业务数据，合并二者，渲染成用户界面，在屏幕上显示出来。GUI-Client 是将界面渲染成通常的窗口界面风格，和 Java 或 .Net 编写的界面程序类似。
- **Web-Client**，这也是一个客户端，不同的是，它把来自数据库的界面和业务数据渲染成 HTML 格式的网页形式，因此你可以通过 IE、Firefox 等浏览器访问。



从这个架构图可以看到，还可以把 Web-Client（从用户角度看，其实是 Web-Server）和 Apache HTTP Server 集成，从而为 Internet 用户提供 Https 的安全访问模式。当需要合作伙伴访问公司的部分数据时，这个功能很有用。

二、 OpenERP 应用服务器架构

OpenERP 技术史上最闻名的恐怕是它的灵活的模块化设计了。它整个应用服务器由松散的模块构成，模块间的耦合度非常低。怎样松散耦合，是软件设计中的一个困难问题，OpenERP 这点在 ERP 软件中做的最好。



OpenERP 的应用服务器有这么几部分构成：

- **ORM**, Object Relation Mapping, 负责数据对象到数据库的访问。在 OpenERP 的业务对象中，你不必写一行数据库访问代码，就自动具备 CRUD 的数据库访问功能。
- **BMD**, Base Module Distribution, 是一个基础模块，必不可少。其他模块，你可以自己任意添加，开源社区有超过 500 个以上的可用模块。
- **Report Engine**, 负责生成各种报表。目前支持的报表格式有 PDF, OpenOffice, HTML 三种。
- **Workflow Engine**, 工作流引擎。支持任意复杂度的工作流，OE 的工作流使用 XML 格式文件定义，目前也提供简单的图形化工作流编辑工具。
- **WebService**, 提供网络调用接口。目前支持 Net-RPC、XML-RPC 两种。

关于 OpenERP 核心对象结构，请参考这里 http://doc.openerp.com/images/pom_3_0_3.png

三、 PostgreSQL 数据库简介

PostgreSQL 是以加州大学伯克利分校计算机系开发的 POSTGRES, 版本 4.2 为基础的对象关系型数据库管理系统 (ORDBMS)。PostgreSQL 被公认为世界上最先进的开放源码的数据库系统，它的许多领先的概念，在晚了好些年后，陆续出现在商业数据库中。PostgreSQL 目前最新的版本是 2009 年 12 月份发布的 8.4.2，它的官方网站在这里：<http://www.postgresql.org/>。

POSTGRES 开发始于 1986 年，它是由加州大学伯克利分校 Michael Stonebraker 领导的，由美国防务高级研究项目局 (DARPA)，美国陆军研究办公室 (ARO)，美国国家科学基金 (NSF)，以及 ESL, Inc 共同赞助的数据库研究项目。POSTGRES 1987 年发布了第一个演示版，并且在 1988 年的 ACM-SIGMOD 大

会上展出。1989 年 6 月发布了版本 1 给一些外部用户使用。为了回应用户对第一版规则系统的批评，作者重新设计了规则系统，并于 1990 年 6 月发布了使用新规则系统的版本 2。版本 3 于 1991 年发布，增加了多存储管理器的支持，并且改进了查询执行器，重写了规则系统。从那以后，随后的版本直到 Postgres95 发布前工作都集中在移植性和可靠性上。1993 年，随着用户数的成倍增涨，Postres 数据库的支持成本越来越大，项目组在发布了 Postres4.2 后终止了项目。

POSTGRES 发布后在许多研究、生产系统中得到了应用。财务数据分析系统，喷气引擎性能监控软件包，小行星跟踪数据库，医疗信息数据库和一些地理信息系统。POSTGRES 还被许多大学用于教学用途。后来，Illustra Information Technologies (后来并入 Informix，后者现在被 IBM 收购了)拿到代码并使之商业化，发布了 Informix 数据库。

1994 年，Andrew Yu 和 Jolly Chen 向 POSTGRES 中增加了 SQL 语言解释器，并优化了全体代码。随后以 Postgres95 的版本号、以开放源码的形式发布到互联网上供大家使用。Postgres95 所有源代码都是完全的 ANSI C。和 Postres4.2 相比，Postgres95 代码量减少了 25%，修正了许多错误，大幅提高了代码性能和代码维护性。Postgres95 版本 1.0.x 进行 Wisconsin Benchmark 测试时大概比 POSTGRES 4.2 快 30-50%。

1996 年，Postgres95 改名为 PostgreSQL，版本定为 6.0。今天，PostgreSQL 的最新版本是 8.4.2。下面列出一些有公开报道的使用 PostgreSQL 数据库的机构。

- **Affymetrix:** (www.affymetrix.com)，一个生物技术研究公司，为基因研究工业提供艺术级的研究工具，在他们的 Transcriptome 项目中用 PostgreSQL 存储那些从高密度 GeneChip 序列中生成的海量 RNA 表达实验。
- **Afilias:** (www.nic.info)，一个域名注册提供商，目前管理互联网上新发布的第一个扩展 .INFO。它是一个位于爱尔兰的公司，管理着超过 900,000 个域名以及超过一千万条记录，这些数据都是存放在以 PostgreSQL 为后端的数据库中。
- **BASF:** (www.basf.com)，巴斯福，世界上最大的化工公司，使用 PostgreSQL 为他们的美国农业综合业的电子商务网站存储数据。
- **Cognitivity:** (<http://www.cognitivity.com/>)，在线学习提供商，它们使用 PostgreSQL 用于他们的表现软件和管理软件。
- **Journyx:** (<http://www.journyx.com>)，世界公认的基于 web 的时间和开销跟踪的服务商，拥有超过一百万最终用户。他们将 PostgreSQL 用于他们所有 unix 平台上的内部数据库操作(Linux, BSD, AIX, Solaris, 以及很快还有 MacOS X)。
- **Royal**，现代 Royal 打字机公司，他们的在线电子商务网站什么东西都用 PostgreSQL 存储。
- **The American Chemical Society**，美国化学社区，(www.acs.org)，世界上最大的专业化学组织，拥有超过 165,000 个成员，以及一个每天有超过一千二百万人访问的网站。他们的刊物档案馆存储了 125 年的所有出版物（两千五百万页，超过 1T 的数据），用的是 PostgreSQL。（目前已知 PostgreSQL 最大的数据库）。
- **Tsutaya**，日本最大的视频节目租赁连锁店，用 PostgreSQL 存储他们超过几个 G，几百万条记录的数据，以及做他们的商业信息数据库。

第三节 OpenERP 功能和应用简介

一、 OpenERP 功能模块

OpenERP 目前的功能模块很多，详细参考这里：<http://www.openerp.com/discover/features.html>。摘录部分如下：

Enterprise Modules（企业管理模块）

- * Sales
- * Purchase

- * Services Management
- * Invoicing
- * Point of Sales

Logistics（物料管理）

- * Warehouse Management
- * Shipping Management
- * Manufacturing
- * Quality & Repairs
- * Products and pricelists

Accounting & Finance（财务管理）

- * Accounting
- * Analytic Accounting
- * Budgets
- * Payments Management
- * Asset Management
- * Bank interfaces

Human Resources（人力资源管理）

- * Expenses
- * Skills Management
- * Holidays
- * Attendances

CRM & SRM（客户和供应商关系管理）

- * Customer Relationship Mgt.
- * Mail Gateway
- * Portals
- * Direct Marketing
- * Phone Calls

Project Management（项目管理）

- * Operational Management
- * Financial Management
- * Timesheets

Daily Productivity（日常工作管理）

- * Integrated DMS
- * Outlook/Thunderbird
- * Getting Things Done
- * Calendars

Efficient Communication（沟通工具）

- * Wiki
- * Webmail

- * Dashboards
- * Alerts

Business Process Management（业务流程管理）

- * End-User Processes
- * Workflow Engine

Association Management

- * Membership
- * Events Organization
- * Fund Raising

IT Companies（IT 公司管理工具）

- * Bug Tracker
- * Scrum Methodology

Fully Customizable（灵活的定制化功能）

- * Report Designer
- * View Editor
- * Workflow Editor
- * Configurable Actions

Ergonomy

- * Web & Application Interfaces
- * Gantt & Calendars
- * Dynamic Graphs
- * Integrated Documentation

Flexible

- * Modules System
- * Web-Services

eCommerce

- * Integrated eCommerce
- * EDI

Business Intelligence

- * Olap Database
- * Cube Designer
- * Data Browser

Miscellaneous Tools

- * Networks
- * Ideas
- * Lunch
- * Voip

二、 OpenERP 的用户群

根据 OpenERP 总部掌握的信息，有据可查的正在使用 OpenERP 的机构超过 250 个，遍布 20 多个国家。根据公开报道，有下面一些著名机构正在使用 OpenERP。

- **法国国家行政学校**(简称 ENA)，孕育法国政治精英与大企业 CEO 的摇篮。从这所学校诞生了两位法国总统，分别是希拉克和德斯坦。ENA 委托 Bull 公司经过详细的市场研究后，选择了 OpenERP 作为学校的全面管理系统。该系统，包括实施 OpenERP 现成功能、定制化开发、和原来系统的集成接口开发，历经 3 年完成。系统用户数 1500 人。
- **比利时商会**(CCI, Chambers of Commerce and Industry of Belgium)，CCI 在比较了 SAP、Navision 后，看中了 OpenERP 无需编程就能实现极其灵活的模块化和个性化配置，从而选择 OpenERP。它们特别看中 OpenERP 的工作流编辑器，视图编辑器，对象编辑器和 Action 编辑器。CCI 实施了成员管理、分支机构管理、事件和培训管理等 13 个模块。项目历时 1 年，用户数 70。
- **卢森堡银行培训机构**(IFBL, Institut de Formation Bancaire Luxembourg)，经历过两个集成项目的失败后，他们选择了 OpenERP，用于计划和培训管理。系统历时 1 年完成，用户数有，培训总部 40 人，银行各分支机构用户约 1000 人。
- **Medical**，medical 是一个基于 OpenERP 平台开发的医院管理信息系统 (HIS) 和电子病例系统 (EMR) 的开源项目，目前已经在阿根廷等国家医院使用。详细信息参考这里：<http://medical.sourceforge.net/>。

中国的 OpenERP 用户群，笔者所掌握的有：

- **上海某二甲医院**，该医院选用 OpenERP 进行 1000 多种物料和医用耗材的库存及科室领用管理。利用 OpenERP 现成的产品、库存、部门、人员等管理功能，自己开发了领料单等功能，在 2 个月之内就完成了系统开发、配置、部署以及培训，正式上线使用。系统使用科室 60，用户数约 200。
- **深圳某顾问公司**，他们选用 OpenERP 管理公司产品、进销存业务、项目、人员、财务等。系统在 3 个月内正式上线使用，目前用户数 17 人。
- **温州某液压厂**，他们选用 OpenERP 管理总部及全国各地的分仓库存。

第四节 OpenERP 开发简介

企业管理软件中，最常见的功能要求是数据访问。如员工信息查询修改，产品订单编辑等。其次是工作流开发。工作流是企业层级协作关系在软件中的体现，如请假单审批，销售订单审批等，在管理软件中都是以工作流来实现。再次是报表开发。系统中的数据，需要以各种形式汇总统计、输出到纸上，送交不同的人或机构。

一、 数据访问功能开发

OpenERP 不仅是一个 ERP 软件，它还是一个高效率的管理功能开发工具。作为 ERP 软件，它提供了很多现成的功能，涵盖标准 ERP 概念中的各项功能。此外，当需要适应企业特殊情况，开发新功能时，OpenERP 非常简单、高效。先说数据访问功能，如产品订单编辑，通常需要一个数据对象，这里是订单，一个视图，即用户界面，还要一个触发动作，通常是菜单或按钮，当用户点击菜单时，系统弹出用户界面，显示、编辑数据。这个方式也就是众所周知的 MVC 模式。OpenERP 的功能开发遵循彻底的 MVC 模式。

数据对象是 Model，在 OpenERP 中，使用 Python 代码实现 Model。如下是一个请假单的代码。

```
class qingjia_qingjd(osv.osv):
    _name = 'qingjia.qingjd'
```

```
_description = '请假单'
_columns = {
    'shenqr': fields.many2one('hr.employee', '申请人', required=True),
    'tians': fields.float('请假天数', required=True),
    'kaisrq': fields.date('开始日期', required=True),
    'shiyou': fields.text('请假事由'),
    'active': fields.boolean('有效'),
    'state': fields.selection([('draft','草稿'),('wait_prove','待批'),('proved','已批'),('rejected','被拒')], '状态',
required=True)
}
qingjia_qingjd()
```

可能你也看出来了，这个代码中其实就是定义了请假单对象的几个字段。在 OpenERP 中，简单的对象，你只要定义对象的各个字段，系统会自动为你创建数据库表，自动生成 CRUD 的数据库操作代码。因此，只要上述代码，你不必另外在数据库中创建 Table，也不必写 Insert、Select、Delete、Update 等数据库操作代码，这些都留给 OpenERP 帮你去搞定。

接下来是用户界面的开发，即 View。OpenERP 中，用户界面的开发不要写任何代码。它是用 XML 格式定义用户界面。如下是请假单查询界面和请假单编辑、审批界面。

请假单

基本查询高级查询

申请人: 开始日期: -

<< 第一页 < 上一页 [0 - 0 共 0] 下一页 > 最后一页 >>

<input type="checkbox"/>	申请人	请假天数	开始日期	请假事由
--------------------------	-----	------	------	------

请假单

查找 表单 日历 甘特图 图形 流程

<< 第一页 < 上一页 [-/0] 下一页 > 最后一页 >>

申请人: 请假天数:

开始日期: 请假事由:

[自定义]

这两个界面，在 OpenERP 中，你只要写如下几行 XML 即可。

```
<record model="ir.ui.view" id="view_qingjd_tree">
    <field name="name">请假单</field>
```

```

<field name="model">qingjia.qingjd</field>
<field name="type">tree</field>
<field name="arch" type="xml">
  <tree string="请假单">
    <field name="shenqr" select="1"/>
    <field name="tians" />
    <field name="kaisrq" select="1"/>
    <field name="shiyou" />
    <field name="state" select="1"/>
  </tree>
</field>
</record>
<!-- 表单视图 代码从略 -->

```

最后是触发菜单的开发，在 OpenERP 中，叫做 Action，也就是 MVC 中的 C，Controller。OpenERP 的 Action 也不用写代码，是用 XML 定义的。



Menu



上述请假菜单，只需下述几行 XML 即可。

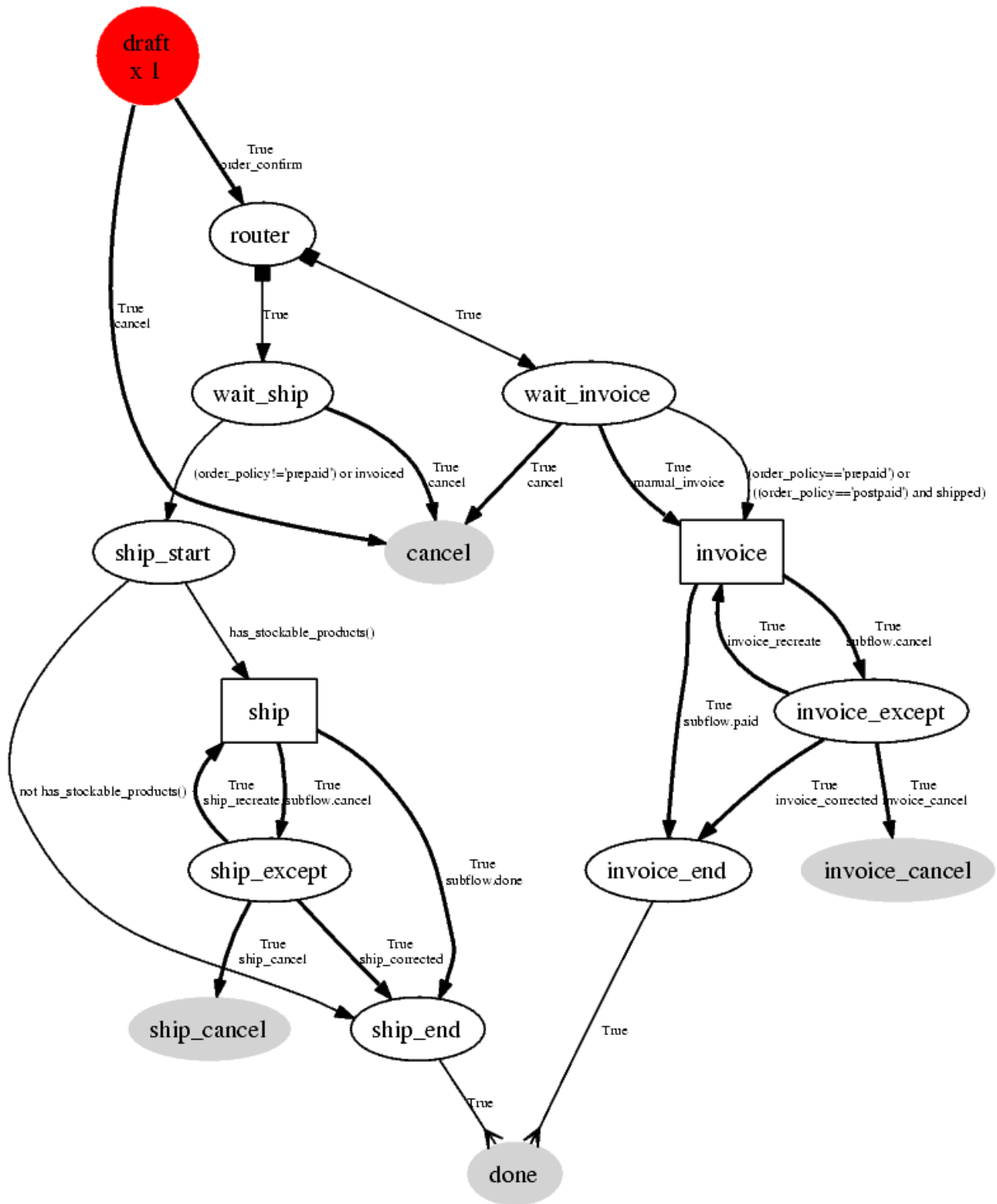
```

<!-- 菜单和 Action -->
<record model="ir.ui.menu" id="menu_qingjia_main">
  <field name="name">请假申请</field>
</record>
<record model="ir.actions.act_window" id="action_qingjia_myqingjd">
  <field name="name">我的请假单</field>
  <field name="res_model">qingjia.qingjd</field>
  <field name="domain">[('shenqr.user_id','=',uid)]</field>
</record>

```

二、 工作流和报表开发

OpenERP 的工作流采用 XML 定义。可以定义工作流的节点，任意复杂的流转条件。如下图的复杂工作流，OpenERP 都能很容易实现。

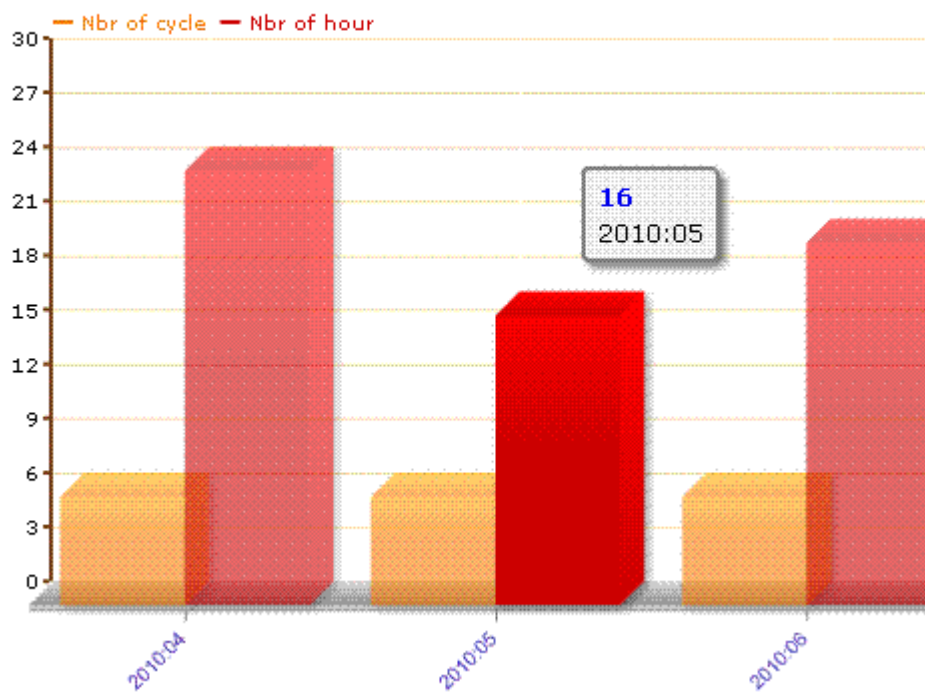


OpenERP 也支持可视化工作流编辑，下图是 OpenERP 的工作流编辑画面。



下图是 OpenERP 的图形化报表。

Workcenter futur load



关于 OpenERP 的详细开发方法，请参看本书后续各章节。

第五节 为什么选择 OpenERP

ERP 软件很多，OpenERP 有什么优势呢。四个字概括，价廉物美。

首先，OpenERP 是开源软件。ERP 软件实施的成本构成中，

业务及需求分析：0 – 10%

软件授权（License）：20 – 30%

系统集成：50% -- 70%

其他：10%

其中，系统集成包括，系统配置，基础数据设置，二次开发，数据恢复，用户培训等。

OpenERP 是开源软件，您可以任意下载源代码，任意修改，不需要授权费。和商业软件相比，您立即节省 20—30% 的费用。

其次，OpenERP 的系统集成成本是所有 ERP 软件中最低的。一是，基于 OpenERP 部署系统就像搭积木一样简单。OpenERP 的模块化是公认最灵活、耦合度最低的。每个功能都是一个独立模块，你可以像搭积木一样，根据功能需要，逐步增加模块。二是，OpenERP 二次开发效率最高。这得益于 OpenERP 是用 Python 语言开发的。OpenERP 是世界上唯一一款用 Python 开发的 ERP 软件。据测算，Python 语言的开发效率是 Java 语言的 5—10 倍。这意味着，同样的二次开发要求，OpenERP 只需要基于 Java 的 ERP 软件的 1/5 甚至 1/10 的开发时间。因此，在 ERP 实施的系统集成阶段，您又可以节约一大笔资金。

第三，丰富的功能模块。OpenERP 有超过 500 个功能模块，涵盖企业管理中的各个方面，涉及各个行业。而且，这些模块都是开源的，您可以任意使用、修改。在全球超过 1000 位开发人员的参与下，模块数还在不断增加中。

第四，全球各行业的案例证明，OpenERP 是稳定的。OpenERP 从 2002 年开始，历经 8 年发展，有报导的客户遍及全球 20 多个国家。大的客户有法国国家行政学校，有 1500 个用户。法国邮政，卢森堡银行，美国最大的白色家电公司惠而浦等。有报导的行业有，金融保险、生产制造、食品、服务业、教育、娱乐、

书店、在线拍卖等。

注：关于 Python 和 Java 开发效率的详细对比，您可以查看这里：

<http://www.developertutorials.com/tutorials/python/python-and-java-a-side-by-side-comparison-8-01-13/page1.html>

第二章 进销存业务概述

第一节 采购业务概述

采购 (purchasing): 是指企业在一定的条件下从供应市场获取产品或服务作为企业资源, 以保证企业生产及经营活动正常开展的一项企业经营活动。根据各行业采购工作的通性和个性, 可将采购管理工作分为四类:

- **生产性采购**, 就是采购这个物品是直接为了公司生产运营所需。而生产性采购又分为两类, 第 1 类是原材料性采购 (MRP 性物料采购)。所采购的此类物料是本企业所生产的产品的组成部分或中间体产品。第 2 类是零配件的采购 (MRO 性采购, 就是维修, 维护、修理、运作), 这些零配件是为了保障机器能正常生产运作所需要的维修、更换配件。
- **商贸性采购**, 像沃尔玛这样的零售商, 它们的采购不属于生产性采购, 属于商贸性采购。商贸性采购和生产性采购最大的区别是, 批发商、零售商在采购物品时, 采购什么商品并不十分重要, 重要的是采购的东西必须保证能赚钱。笔笔都得赚钱, 不赚钱就没有必要采购进货。西瓜不赚钱, 可以采购苹果。但是对于生产性采购来讲就不行, 这个产品我们公司只能用这个原材料。我做饼干的, 只能买面粉, 不能说面粉太贵就不买了, 不买就停产了。所以生产性并不保证每次采购都赚钱, 而且采购的盈利性也不能直接反应出来的。
- **一般日常用品性采购**, 像办公用品采购, 还有行政采购等。其特点是采购品类繁杂, 但采购金额小。其所采购的物品主要是保障公司的正常行政办公所用。
- **项目性采购**, 有时候, 我们的采购工作属于项目性采购, 比如说买一台设备, 盖一个车间等。项目性采购的主要特点是一次性。很少有重复性的采购。这就意味着每次采购的流程都得重新开始, 以往的经验 and 关系很少能用到。

一般的 ERP 软件, 都只支持生产性采购中的 MRP 性物料采购, 专业采购软件, 可能支持所有类型的采购业务。OpenERP 的采购, 主要也是 MRP 性物料采购。不过, 可能存在其它模块支持别的采购管理, 如办公用品的请领、采购。本书的采购, 如不特别说明, 指的是 MRP 性物料采购。

采购的日常工作有, 接收采购计划、询价/议价、决定、下 PO (采购单)、审核、跟催、收货、付款、退货。相关的单据有: 请购单、采购计划、采购单、询价单, 进货单等。简单的说, 采购部门接收其它各部门的采购要求, 定期汇总采购物品, 作成采购计划。根据采购计划, 分别和相应的供应商询价、议价。向选定的供应商下单 (通常是传真, 供应商签字后再回传回来)。而后跟踪供应商及时发货, 货到后验货、入库。货有问题的话, 要求供应商换货、退货。

采购的财务处理, 供应商的发票送至财务部门。采购部门向财务部门申请付款, 或者出具供应商签字确认过的采购单复印件至财务部门, 代替付款申请。仓库收货后将入库单亦送至财务部门。财务部门凭此三单 (采购部门的付款申请、供应商发票、仓库的入库单), 核对三单一致后, 记录会计帐务, 制定付款计划, 按计划付款。

当采用 OpenERP 后, 典型的采购流程大致是这样的。仓管员分析库存, 对于低于库存警戒线的物品, 提出采购申请 (系统里叫 Procurement Orders)。或者, 如果库存管理模块中配置了最小库存规则, 那么, 系统每天自动计算, 低于最小库存的产品, 系统自动产生采购申请。如果是自产的产品, 不是外购的产品, 则系统自动生成生产单, 生产单根据 BOM 计算所需物料, 如果物料不足, 系统再生成采购申请。采购员汇集采购申请 (通常是将同一供应商的采购申请合并成一张采购订单); 或者, 直接在系统中新建一张采购单, 输入要采购的物品的规格型号、价格等, 而后和供应商 Email 或电话沟通, 进一步敲定价格等。采购员提交采购单, 经理审批。审批后, 采购员打印采购单, 传真给供应商, 供应商签字确认, 回传。采购员点击采购单上的“供应商通过”按钮, 表示供应商已确认。此时, 系统自动生成 2 张单子, 一是供应商

凭证 (Invoice), 送至财务部门, 财务部门据此审核供应商发票并安排付款计划。当财务收到供应商发票, 确认供应商凭证后, 系统会自动生成此采购业务对应的记账凭证。二是收货单 (Incoming Packing List), 送至仓管部门, 仓管部门据此验货、收货。

付款并核销的处理。当出纳付完采购款项后, 可以在 OpenERP 中输入付款记录, 并选择要冲减的采购凭证。这样, 系统会自动生成付款会计分录, 并自动核销 (Reconcile) 应付款。应付款核销后, 显示在业务伙伴画面的对应供应商的应付款项会被冲减。采购订单上也会标记该订单已付款。

关于预付款的处理。有些采购要求预付部分款项, 此时, 采购部门要向财务部门出具预付申请单。在 OpenERP 软件中, 目前应该还不支持预付申请。作为实践中的补充措施, 可以有两种处理办法, 一种是手工的, 即手工填写预付申请单, 交由财务, 财务据此付款, 并手工录入预付款的记账凭证。二是使用电子邮件。预付申请是一种内部单据, 可以使用企业内部电子邮件作为内部单据的高效流转方式。采购员发送预付申请邮件给主管和财务, 主管回复邮件确认, 而后财务也回复邮件确认。财务打印最后的确认邮件, 作为内部单据原始凭证, 以此记账。这种方式, 首先需要严格管理邮件, 保证电子邮件的安全, 其次, 企业的财务制度要承认电子邮件作为内部凭证的有效性。

第二节 销售业务概述

企业的销售工作是非常重要的, 销售旺盛, 才能带动其他部门和工作的快速发展。企业的销售工作, 通常分成两大块, 营销管理和销售管理。营销管理是为了实现各种组织目标, 创造、建立和保持与目标市场之间的有益交换和联系而设计的方案的分析、计划、执行和控制。销售管理是计划、执行及控制企业的销售活动, 以达到企业的销售目标。简而言之, 营销是在企业和目标市场间营造一个良好的交换气氛, 销售是在企业和目标市场间完成具体的产品交换。

营销管理的工作有: (1)发现和评价市场机会; (2)细分市场和选择目标市场; (3)发展市场营销组合和决定市场营销预算; (4)执行和控制市场营销计划。通常, 在 ERP 软件中, 对营销工作的支持是 CRM (客户关系管理) 模块的功能。

销售管理的工作有: (1)制定销售计划及相应的销售策略; (2)建立销售组织并对销售人员进行培训; (3)制定销售人员的个人销售指标, 将销售计划转化为销售业绩; (4)对销售计划的成效及销售人员的工作表现进行评估。在 ERP 软件中, 对销售管理工作的支持, 通常也是 CRM 模块的范畴。ERP 软件中的销售模块, 通常是指, 对具体的销售业务工作的支持, 即报价、成交、发货、收款、记账的具体业务工作的支持。

ERP 软件里的销售, 通常是指面向机构的销售, 而不是面向普通消费者的零售。零售在 ERP 软件中, 通常是 POS (Point Of Sale) 模块的功能。面向机构的销售业务的基本过程是: 挂牌标价→洽谈业务→签订合同→收取货款和发运商品→开具发票及提单→提供售后服务和客户回访。

- **挂牌标价。**企业向外发布产品价格信息。
- **洽谈业务。**挂牌标价后, 企业销售人员就开始和有购买意向的客户进行交易条件的洽谈。
- **签订合同,**销售订单是销售合同的一种。
- **收取货款和发运商品。**
- **开具增值税专用发票。**
- **开具发货单。**发货单是购货方提取商品的凭证, 也是供货方开展销售业务的内部凭证。
- **提供售后服务和客户回访。**商品售出以后, 企业应根据具体情况提供良好的售后服务, 如送货上门、安装调试、维修保养、使用指导等等。除此之外, 还应定期回访, 听取顾客对产品及服务的建议和意见, 以便更好地改进产品、完善服务。在软件中, 这部分销售相关的内容通常属于 CRM (客户关系管理) 的范畴。

当基于 OpenERP 处理销售业务时, 大致流程是这样的。销售人员收到客户购买意向后, 通常是在系统中新建一张报价单, 将客户需要的产品型号、价格、付款条件、发运方式等信息录入系统。打印或 Email 报价单给客户, 要求客户确认。如果是新客户, 可能还要销售经理、财务等审核客户信誉、订单风险等问

题。系统目前不支持新客户及订单风险的审核流程。作为系统补充，可以采用手工审核的方式。当和客户敲定订单相关事宜，公司内部各项审核要求也通过后，确认报价单，报价单就变成了销售订单。这时，系统会自动生成一张销售凭证（Customer Invoice），送交财务部门，财务部门根据该凭证开发票及记账。系统还会自动生成出货单，送交仓库，仓库凭此出货。如果销售订单上有些产品是“按订单生产”（即有订单才启动生产，对应的，另一种方式是“按库存生产”，即不管有无订单，都生产，产成品存入仓库。），则系统会自动生成该产品的生产单（Product Order）。如果需要更复杂的销售订单处理流程，例如，超过一定金额的要销售主管或者总经理审核，则必须定制销售订单的工作流。这可以在“系统管理”菜单下修改销售订单的工作流，或者直接修改代码中的工作流定义的 XML 文件。

销售业务中的单据主要有销售订单、增值税专用发票和发货单。增值税专用发票的内容有，购货方单位名称、纳税人登记号、开户银行及账号等。增值税专用发票的联次及其用途：

- 第一联为存根联，由销货方留存备查。销货方在使用专用发票时，必须按规定保持各联号码连续完整，不得从专用发票本上撕下。
- 第二联为发票联，供购货方作记账使用，只属于商事凭证。
- 第三联为税款抵扣联，是购货方计算进项税额的证明，也是向税款征收机关进行税款抵扣的依据，由购货方持有。
- 第四联为记账联，是销货方核算销售额和销项税额的主要依据。

下图为增值税发票样例。

3100043140 上海增值税专用发票 No 13428210 开票日期：2005年03月29日

购 名 称： 华明(上海)有限公司		密 码： */17702/>+<+05>43263	
纳 税 人 识 别 号： 310227734547339		/ >063/+2+*93<0041-2<8	
地 址、电 话： 松江区阳光镇 57774835		9>8-032+39/-3<+<0*1/0	
开 户 行 及 帐 号： 中国银行松江支行 049400-8350-101008091001		7<16570>>2-*84/8->>49	
货 物 或 应 税 劳 务 名 称： 漆棉细布 45"		规 格 型 号： 21*21 60*60	单 位： 百米
数 量： 15		单 价： 213.6752	全 额： 3205.13
税 率： 17%		税 额： 544.87	
合 计		¥3205.13	
价 税 合 计 (大 写)： 叁仟柒佰伍拾圆整		(小 写)： ¥3750.00	
销 名 称： 华光 股份有限公司		备 注： 0805	
纳 税 人 识 别 号： 310115132212080			
地 址、电 话： 上海市浦东新区陆家嘴路166号,50624567			
开 户 行 及 帐 号： 中行银行上海市分行044036-8001-00702408091001			
收 款 人：		复 核：	开 票 人： 方艳
		销 货 单 位： (章)	

国税通[2004]683号上海印钞厂 第一联：抵扣联 购货方扣税凭证

发货单是购货方提取商品的凭证，也是供货方开展销售业务的内部凭证。发货单各联次及其用途如下：

- 第一联为记卡联，销售员记卡留存。
- 第二联为提货联，购货方凭此提货。
- 第三联为收款联，财务凭此收款记账。

第三节 采购、销售的财务处理

先说采购记账。例如北京鹏瑞公司，是一商贸类型的公司，他们从北京维佳科技公司购入 A 产品 100

件，单价 1500，单价未含税，增值税 17%。那么，对应这笔业务，会计上通常做如下会计分录：

```
借: 1402 在途物资      150000
    2015 应交税费 - 201501 应交增值税(进项税)    25500
    贷: 2011 应付帐款 - 201101 维佳科技公司    175500
```

采购商品入库时，会计分录：

```
借: 1036 库存商品      150000
    贷: 1402 在途物资    150000
```

根据付款条件，财务编制付款计划。到了付款日，出纳开具转帐支票去银行付款。根据转帐支票的留底，会计上记账如下：

```
借: 2011 应付帐款 - 201101 维佳科技公司    175500
    贷: 1002 银行存款    175500
```

采用 OpenERP 后，确认采购单时，系统会自动生成采购凭证。财务部门确认采购凭证时，系统自动生成上述记账凭证（根据设置的不同，也可以不自动生成，而是手工录入）。采购商品入库时（确认入库单时），系统自动生成入库会计分录。出纳付完款后，可以手工录入付款凭证。也可以利用系统的对账单功能，输入付款额，选择未支付发票（Invoice），冲销（Reconcile）发票上的应付账款，系统自动生成付款的记账凭证。

再说销售记账。鹏瑞公司向上海福达公司销售 A 产品 10 件，单价 2000。对应这笔业务，会计上记账如下：

```
借: 1014 应收账款 - 101401 上海福达公司    21700
    贷: 6001 主营业务收入                    20000
        2015 应交税费 - 201501 应交增值税(销项税)    1700
```

销售商品出库时，销售实际成本发生，必须结转销售成本，记账如下。传统手工记账时，销售成本的结算通常是月末时进行。使用 OpenERP 的话，商品出库时（出库单确认时），系统自动生成销售成本结算分录。

```
借: 6015 主营业务成本    15000
    贷: 1036 库存商品      15000
```

当对方付款后，根据银行回单，会计上记账如下：

```
借: 1002 银行存款      21700
    贷: 1014 应收账款 - 101401 上海福达公司    21700
```

采用 OpenERP 后，确认销售订单时，系统会自动生成销售凭证。财务部门确认销售凭证时，系统自动生成上述记账凭证（根据设置的不同，也可以不自动生成，而是手工录入）。出纳收到款后，可以手工录入收款凭证。也可以利用系统的对账单功能，录入收款额，选择应收发票（Invoice），冲销（Reconcile）发票上的应收账款，系统自动生成收款的记账凭证。

如果采用 ERP 软件处理购销业务中的财务管理，首先必须建立会计科目表（Chart Of Account）。上述简单的购销业务，需要建立下述科目表：

科目代码	科目名称
1001	库存现金
1002	银行存款
1036	库存商品

1014	应收账款
101401	上海福达公司
2011	应付账款
201101	维佳科技公司
2015	应交税费
201501	应交增值税(销项税)
201502	应交增值税(进项税)
6001	主营业务收入

在后续章节中，将会介绍怎么在 OpenERP 中设置科目表，怎么设置业务伙伴、产品，以及怎么完成采购、销售业务，购销业务怎么自动财务处理。

第四节 仓库管理业务概述

仓库管理也叫仓储管理，英文 Warehouse Management，简称 WM，指的是对仓储货物的收发、结存等活动的有效控制。其目的是为企业保证仓储货物的完好无损，确保生产经营活动的正常进行，并在此基础上对各类货物的活动状况进行分类记录，以明确的图表方式表达仓储货物在数量、品质方面的状况，以及目前所在的地理位置、部门、订单归属和仓储分散程度等情况的综合管理形式。注意仓库管理和库存管理的区别。库存管理，英文 Inventory Management，是根据外界对库存的要求、企业订购的特点，预测、计划和执行一种补充库存的行为，并对这种行为进行控制，重点在于确定如何订货，订购多少，何时定货。库存管理的主要工作是规划和保证最合适的库存。仓库管理的主要工作是处理物品入库、出库以及在库保管等具体的业务工作。

仓库管理的基本业务包括：商品入库、商品出库、库房管理等。库房的管理，要根据商品类型的不同，将仓库分成若干库区（Location），分区管理。例如，合格品区、收货区、辅助区、不合格品区。商品储存货架要设置存货卡，商品进出要遵循先进先出的原则。

商品入库是商品进入仓储时所进行的商品接收、卸货、搬运、清点数量、检查质量和办理入库手续等一系列活动的总称。商品入库作业包括商品接运、商品验收、办理入库交接手续等一系列业务活动。商品接运是指仓库对于通过铁路、水运、公路、航空等方式运达的商品，进行接收和提取的工作。商品验收，包括数量点收和质量检验两方面的验收。入库手续主要是指交货单位与库管员之间所办理的交接工作。包括商品的检查核对，事故的分析、判定，双方认定，在交库单上签字。仓库一面给交货单位签发接收凭证；一面登记入库，并将入库凭证交给会计入账。

商品出库业务，是仓库根据业务部门或存货单位开出的商品出库凭证（提货单、调拨单），按其所列商品编号、名称、规格、型号、数量等项目，组织商品出库一系列工作的总称。商品出库的形式有：

- **送货**，仓库根据货主单位预先送来的“商品调拨通知单”，通过发货作业，把应发商品交由运输部门送达收货单位，这种发货形式就是通常所说的送货制。
- **自提**，由收货人或其代理持“商品调拨通知单”直接到库提取，仓库凭单发货，这种发货形式就是仓库通常所说的提货制。它具有“提单到库，随到随发，自提自运”的特点。
- **过户**，是一种就地划拨的形式，商品虽未出库，但是所有权已从原存货户转移到新存货户。仓库必须根据原存货单位开出的正式过户凭证，才予办理过户手续。
- **取样**，货主单位出于对商品质量检验、样品陈列等需要，到仓库提取货样。仓库也必须根据正式取样凭证才予发给样品，并做好账务记载。
- **转仓**，货主单位为了业务方便或改变储存条件，需要将某批库存商品自甲库转移到乙库，这就是转仓的发货形式。仓库也必须根据货主单位开出的正式转仓单，才予办理转仓手续。

商品出库程序包括核单备料→复核→包装→点交→登账→现场和档案的清理过程。

- **核单备料**，如属自提商品，须检查有无财务部门准许发货的签章。

- **复核**，复核内容包括品种数量是否准确，商品质量是否完好，配套是否齐全，技术证件是否齐备，外观质量和包装是否完好，等等。复核后保管员和复核员应在“商品调拨通知单”上签名。
- **包装**，要写明收货单位、到站、发货号、本批总件数、发货单位等。
- **点交**，商品经复核后，如果是本单位内部领料，则将商品和单据当面点交给提货人，办清交接手续；如系送料或将商品调出本单位办理托运的，则与送料人员或运输部门办理交接手续，当面将商品交点清楚。交清后，提货人员应在出库凭证上签章。
- **登帐**，保管员应在出库单上填写实发数、发货日期等内容，并签名。然后将出库单连同有关证件资料，及时交给货主，以使货主办理货款结算。保管员把留存的一联出库凭证交给实物明细账登记人员登记做账。
- **现场清理**，包括清理库存商品、库房、场地、设备和工具等；档案清理是指对收发、保养、盈亏数量和库位、垛位安排等情况进行分析。

仓库管理的常见单据有，商品入库单、商品送货单、物料入库单、物料出库单、物料盘点表、物料退仓单、借物登记表。

第三章 OpenERP 基础业务设置

要使用信息系统管理进（采购）、销（销售）、存（库存管理）业务，首先要将业务系统中的各种管理对象数字化。在进销存中，基础的业务对象包括业务伙伴、产品、仓库、财务科目等。例如，销售业务中，信息系统要生成一张销售订单，销售订单上包括客户信息（卖给谁、送货地址等）、产品信息（销售什么产品，什么型号，什么价格等）、仓库信息（从哪个仓库出货）、财务信息（本笔业务借记什么、贷记什么，应收款多少等）。必须事先将业务对象数字化，信息系统才能快速、简单的支持业务操作，管理业务过程。

- **业务伙伴**：向谁采购，销售给谁，这些业务伙伴的信息要预先设置好。
- **产品**：采购什么，销售什么，这些产品信息要预先设置好。
- **仓库**：当有多个仓库，每个仓库还分多个库区时，不同类型的产品存放在哪个库区，销售时从哪个仓库发货。更智能一点，库存低于多少时希望系统自动提醒补货。这些仓库相关信息要预先设置好。
- **财务**：采购、销售都需要财务记账，需要资金的收支管理。财务相关信息要预先设置好，系统才能记账。

目前 OpenERP 的界面汉化存在多个版本，一个是官方的 5.07 版的汉化，一个是 HornERP (<http://code.google.com/p/hornerp/>) 对 5.06 版的汉化。两个版本术语不尽一致，且两个版本都存在一些翻译欠准确的术语。为了减少术语翻译带来的理解困难，本章采用英文界面介绍。

第一节 业务伙伴信息设置

OpenERP 的业务伙伴，包括任何和自己有关的单位、个人。可以是客户，也可以是供应商。一个业务伙伴可以有多个地址，如联络地址、发票地址、送货地址、各办公场所的地址等。如果安装了 `base_contact` 模块，每个地址可以有多个工作职位及对应的联系人。每个联系人可以有多个工作职位、从属于多个业务伙伴，甚至不属于任何业务伙伴（私人地址）。

一、 理解业务伙伴

业务伙伴和联系地址。业务伙伴的主要信息存放在数据表 `res_partner` 里，一个业务伙伴有多个联系地

址，联系地址信息保存在数据表 `res_partner_address` 里。`res_partner` 和 `res_partner_address` 是 One2Many 关系。

业务伙伴的价格表和库位属性。可以为每一个业务伙伴设定一个采购价格表和销售价格表。价格表定义专属于该业务伙伴的产品价格，通常是基于产品公开报价的折扣率。稍有遗憾的是，OpenERP 只能针对单个业务伙伴分别设定价格表，不能针对某一类业务伙伴设定价格表。业务伙伴的库位属性，当从供应商进货或销售给客户时，OpenERP 不是直接增加或减少库存数量，而是记录从供应商库位到公司库位的库存移动，以及从公司库位到客户库位的库存移动。业务伙伴的库位属性就是设定此处的供应商库位和客户库位，可以是虚拟的库位，也可以是供应商或客户的真实库位。

业务伙伴的财务属性。当和业务伙伴发生交易时，会产生应收、应付款，应收应付款在财务上记录在哪个会计科目上？这就是业务伙伴的财务属性设置。此外，业务伙伴的财务属性中还会显示业务伙伴的应收应付总额，银行帐号等信息。

二、 业务伙伴联系信息

在 General 页，包括业务伙伴的联系人，地址，电话等信息。

- **Customer:** 如果勾上，表示该业务伙伴是客户，在销售订单上允许选择该客户。
- **Supplier:** 如果勾上，表示该业务伙伴是供应商，在采购订单上允许选择该客户。
- **Title:** 业务伙伴头衔，诸如公司、医院、区政府等。头衔在业务伙伴的“Configuration”菜单中设置。在发送给业务伙伴的信函或订单中会出现该头衔。
- **Language:** 语言，会影响发给客户的订单上的显示语言。国内伙伴，一般都设成“中文”。
- **Type:** 联系人的头衔，诸如先生、小姐、女士等。在“Configuration”菜单中设置。
- **Function:** 联系人职位，诸如总经理、销售员等。在“Configuration”菜单中设置。
- **Address Type:** 地址类型，诸如送货地址，发票地址等。有 5 种地址类型：(default,Default)，(invoice,Invoice)，(delivery,Delivery)，(contact,Contact)，(other,Other)。Default，默认，当没有其他地址时使用此地址。Invoice，发票，系统生成采购、销售等的财务票据时，使用此地址。Delivery，发货，系统生成送货单时，使用此地址。Contact、Other，供其他用处，系统中未使用。当系统生成单据时，会自动使用相应类型的地址。
- **Categories:** 分类，一个业务伙伴可以属于多个分类。在“Configuration”菜单中设置。分类可以是树状结构，如下图所示：

Customer / Segmentation / Bad customers
Customer / Segmentation / Important customers
Supplier / Components Supplier
Supplier / Open Source Service Company
Supplier / Textile Suppliers

业务伙伴联系信息画面。

Partners Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [2/3] Next > Last >>

Name: 北京鹏瑞公司 Code: ? Customer: ☒
 Title: ? Language: ? Supplier: ☐

General Sales & Purchases History Notes Accounting

Partner Contacts [1/1]

Contact Name: 北京鹏瑞公司 Type:
 Function: Address Type: Default
 Street: 某地址 Street2:
 Zip: 1367 City: 某城市
 Country: 中华人民共和国 Fed. State: 北京市
 Phone: (+32). 81. 81. 37. 00 Fax:
 Mobile: E-Mail:

Categories

<< First < Previous [0 - 0 of 0] Next > Last >>

☐ Full Name

REPORTS

- Overdue Payments
- Labels

ACTIONS

- Mass Mailing
- Send SMS
- Company Architecture

LINKS

- Sales
- Purchase orders
- Invoices
- All account entries
- Receivables & Payabl
- Events

三、 业务伙伴购销信息

购销信息在“Sales & Purchases”页面。

General Sales & Purchases History Notes Accounting

General Information

? Dedicated Salesman: Active: ☒
 Website: Date:
 Main Company:

Sales Properties **Purchases Properties**

? Sale Pricelist: Public Pricelist (元) ? Purchase Pricelist: Default I

Stock Properties

? Customer Location: Customers
 ? Supplier Location: Suppliers

[CUSTOMISE]

- **Dedicated Salesman:** 专任销售人员。你可以安排专门销售员跟踪该客户。
- **Main Company:** 业务伙伴所属母公司。
- **Sale Pricelist:** 针对该客户的销售用价格表。价格表处理折扣问题，如某消暑产品，针对某大客户，在公开报价基础上，夏秋两季打 8.5 折，冬春两季打 6 折。当设置好该价格表，销售订单中，将根据价格表中的规则自动计算给该客户的价格。价格表在产品模块中设置。
- **Purchase Pricelist:** 针对该供应商的采购用价格表。当设置好该价格表，采购订单中，将根据价格表中的规则自动计算产品的采购价格。
- **Customer Location:** 客户库区，销售给该客户的产品应发往的库区。可以是真实库区，也可以是一个虚拟库区。OpenERP 采用独特的“复式库存”管理库存，因此，即使客户没有真实库区，也必须设置一个虚拟库区。库区（Location）在库存管理模块中设置。
- **Supplier Location:** 供应商库区，概念同上。

四、 业务伙伴事件

销售人员跟踪客户的过程中，会经常和客户联系。或者拜访、或者电话、或者 Email。和客户联系时，会发现客户的某些潜在需求，或是客户不满。和客户的这种沟通以及客户的需求或者态度变化，在 OpenERP 中使用事件（Event）来记录管理。当销售订单确认时，系统也会自动生成该业务伙伴的“购买事件”记录于此。

General	Sales & Purchases	History	Notes	Accounting
Partner Events				
<< First < Previous [1 - 1 of 1] Next > Last >>				
Date	Events	State of Mind	Partner Relation	User
03/12/2010 13:20:05	投诉电话	不满	Customer	系统管理员
Import Export << First < Previous [1 - 1 of 1] Next > Last >>				

- **Date:** 事件发生的日期
- **Events:** 事件标题
- **State Of Mind:** 客户态度。客户态度的选项在“Configuration”菜单中设置。
- **Partner Relation:** 客户关系类型。
- **User:** 跟踪该事件的职员。

点击事件，进入事件编辑画面，可以记录事件的详细信息。

五、 业务伙伴财务信息

- **Account Receivable:** 和客户发生的销售业务记账凭证中，应收账款对应的会计科目。通常是“1014 应收账款”总账科目下的以客户单位名称为名的二级科目。该科目在财务模块中设置。简单情况下，也

可以将所有客户都设成“1014 应收账款”科目，设置方法在“管理员”下的“属性”菜单中设置属性。销售业务中，系统会根据这里的设置自动生成会计分录。

- **Account Payable:** 和业务伙伴发生的采购业务记账凭证中，应付账款对应的会计科目。通常是“2011 应付账款”总账科目下的以客户单位名称为名的二级科目。
- **Fiscal Position:** 应税设定。通常，在产品信息中设定销售、采购该产品应缴纳的税种。之后，在销售订单、采购订单中，系统会自动采用产品上设定的税种计税。但是，有时候，同一产品，针对不同类型客户，其计税方式不同。例如，某产品内销时要计消费税，但外销时不计消费税。那么，对于外销客户，应该采用不同于产品上设定的计税方法计税。**Fiscal Position** 就是处理这种情况的。当指定客户的应税设定后，系统会根据应税设定中指定的替换规则将产品中设定的税种换成别的税种计算税额。应税设定在财务模块中设置。
- **Payment Term:** 付款条件，如 30 日内付全款，或者 10 日内付 30%，20 日内再付 30%，余款 2 月内付清。当设置好付款方式后，如该客户未按时付款，系统会自动报警。付款条件在财务模块中设置。
- **Total Receivable:** 该业务伙伴的应收账款总额。
- **Total Payable:** 该业务伙伴的应付账款的总额。
- **Credit Limit:** 该业务伙伴的信用额度。
- **Bank Details:** 该业务伙伴银行帐号相关信息，用于增税发票等单据。

GeneralSales & PurchasesHistoryNotesAccounting

Customer Accounting Properties

? Account Receivable: 101401_? Fiscal Position: ? Payment Term:

Supplier Accounting Properties

? Account Payable: 201102 上海

Customer Credit

? Total Receivable: 0.00Credit Limit: 0.00

Supplier Debit

? Total Payable: 0.00

Bank Details

<< First < Previous [0 - 0 of 0] Next > Last >>

Bank Type	Account Owner	Account Number
-----------	---------------	----------------

第二节 产品信息设置

一、 产品模块配置

产品模块安装时的配置步骤，有一个”Configure Sale Order Logistic”（配置销售订单物流方式）的画面。说明如下。

- **Packing Default Policy:** 有两个选项：Direct Deliverry、All at Once。这个选项影响销售订单的发货策略。Direct Deliverry 使得只要仓库有货，即使不足订单要求的量，也立即发货（允许部分多次发货）。All at Once 使得，仓库中必须有足够订单要求的量，才能发货（一次性足额发货）。这里的选项只是

销售订单的默认方式，在销售订单上还可以修改。

- **Shipping Default Policy:** 有两个选项：Invoice Based on Deliveries、Invoice Based on Sales Orders。即根据发货单开票和根据订单开票。前者，当一张订单多次分批发货时，会开出多张发票。后者则不管分几次发货，一张订单一张发票。这里的设置只是销售订单的默认方式，在销售订单上还可以修改。
- **Steps To Deliver a Sale Order:** 有两个选项：Delivery Order Only、Packing List & Delivery Order。意思不是很明白，参看下述英文解释。 By default, Open ERP is able to manage complex routing and paths of products in your warehouse and partner locations. This will configure the most common and simple methods to deliver products to the customer in one or two operations by the worker.

产品单位的配置。菜单 Products → Configuration → Units of Measure → Units of Measure，画面如下。

- **Name:** 单位名称，如个、箱、盒等。
- **UoM Category:** 单位分类，如重量单位，长度单位，PCB 板转换单位等。同一分类的单位间可以互相转换，不同分类的单位间不能转换。单位分类在菜单 Products → Configuration → Units of Measure → Units of Measure Categories 中设置。
- **Factor 和 Rate:** 1 本单位 = 1 基本单位 * factor。如 1 箱=40 个，则可以定义“个”为基本单位，“个”的 factor=1，“箱”的 factor=40。Rate=1/Factor。
- **Rounding Precision:** 单位换算时的精确度，单位换算后的结果必须是此值的倍数。

价格表（Pricelist）的配置，菜单 Products → Pricelists → Pricelists。

Pricelist Version

Name: ? Active: ☒

? Start Date: ? End Date:

Products Listprices Items

Sequence	Rule Name	Product Template	Product Category	Min. Based Quantity on
5	Public	Pricelist Line		0 Public Price

- **Start Date 和 End Date:** 本价格表生效的时间范围。
- **Products Listprices Items:** 价格计算规则，一个价格表可以有多条规则。系统计算产品价格时，按着 Sequence 从小至大依次检查各个规则，如果符合规则条件，则适用该规则计算价格。规则设置画面如下。

Products Listprices Items

Rules Test Match

? Rule Name:

? Product: ? Product Template:

? Product Category:

? Min. Quantity: Sequence:

Price Computation

? Based on: If Other Pricelist:

New Price = Base Price * (1 +) +

? Rounding Method: Min. Margin: Max. Margin:

- **Rules Test Match:** 规则适用条件，只有符合本条件，才会使用本规则计算价格。此处的多个条件是 AND 关系。
- **Product 和 Product Template:** Product 对象继承自 Product Template，目前两者实质是一样的。如果设置了，表示只有该产品适用本价格计算公式。
- **Product Category:** 如果设置了，表示该产品分类适用本价格计算公式。
- **Min. Quantity:** 只有购销数量大于本设置时才适用本价格计算公式。
- **Sequence:** 规则序号，系统优先适用序号小的规则。
- **Price Computation:** 价格计算公式。
- **Based on:** 价格计算基础值，有 Public Price, Cost Price, Other Pricelist, Partner section of the product form 等选项。Other Pricelist 表示，基于别的价格计算规则算出的值的基础上，再适用本规则的公式。这使

得多个价格表可以串联使用。Partner section of the product form 表示，基于产品表单的 Supplier Information 中设置的单价计算。

- **New Price:** 价格计算公式是 $\text{New Price} = \text{Base Price} * (1 + X) + Y$ 。
- **Rounding Method:** 价格精度，计算出来的价格自动舍入变成此值的倍数。例如，如果设成 0.05 则计算所得价格为 45.66 的话，系统会自动转换价格成 45.65；如果设成 100，价格 14567 会转换成 14600。
- **Min. Margin 和 Max. Margin:** 利润范围，如果 $\text{min}=10, \text{Max}=50$ ，则新价格必须满足： $\text{Base Price} + 10 < \text{New Price} < \text{Base Price} + 50$ ，不在此范围的话，系统自动取该范围的上下限作为新价格。

二、 产品一般信息

- **Can be sold:** 该产品可用于销售，在销售订单中可以选择该产品。
- **Can be Purchased:** 该产品可以采购，在采购订单中可以选择该产品。
- **Rentable Product:** 该产品可用于出租。
- **Variants:** 产品型号，任意文字。
- **EAN13:** 产品条形码，必须符合条形码格式。
- **Product Type:** 有三个选项：Stockable Product、Consumable、Service。Stockable Product 是通常的可库存的产品，要求进行库存管理。Consumable 是低值易耗品，如螺丝钉。低值易耗品可库存，但系统不记录其库存量，不对它做库存管理（默认为取之不竭的东西）。Service，服务类型的产品，如咨询服务。
- **Procure Method:** 生产类型，有 Make to Stock、Make to Order 两个选项。即按库存生产和按订单生产。属于后者的产品，当有订单时，会根据订单和产品 BOM 自动生成生产订单或采购订单（取决于产品的 Supply Method），启动产品供应。前者则是当库存低于设定的量时才启动供应。如果是 Service 类型的产品，且设定为 Make to Order，那么，当有订单时，系统会自动生成该服务项目对应的 Task（任务，这个时候，产品的 BOM 相当于是项目的 WBS，工作分解结构）。
- **Supply method:** 供应方式，有两个选项：Produce、Buy。前者的话，当需要补给该产品时（例如库存低于设定量），系统自动生成生产订单。后者的话，当需要补给时，系统自动生成采购订单。
- **Weights:** 记录产品的体积、毛重、净重信息。
- **Category:** 产品分类。产品分类在“Configuration”菜单中设置。产品分类可以形成树状结构。
- **Status:** 产品所处生命周期阶段。有 In Development、In Production、End of Lifecycle、Obsolete 几个选项。
- **Product Manager:** 产品经理。

- **Real Stock:** 产品实际库存，即目前仓库（包括所有仓库）保有的产品实际数量。
- **Virtual Stock:** 产品预期库存。由于有一些已经确认了的采购订单和销售订单，随着这些订单的入库、出库，产品的将来库存量会不同。在库存管理模块中，可以查看产品 N 天后的预期库存。这里是考虑所有已确认的订单而计算的预期库存，不特指哪一天。
- **Lots:** 生产批号和追踪批号。可以强制要求产品必须有生产批号、进货批号、出货批号。批号在库存模块中设置。
- **UOM:** 产品计量单位。产品可能有多个单位，如电池产品，进货时按“箱”计、销售时按“个”和“盒”计。产品计量单位在“Configuration”菜单中设置。OpenERP 中的产品计量单位可分类，同一类型的计量单位间可互相换算（不同类型的单位间不可换算）。如电池的三个计量单位（箱、盒、个）可以设成同一类型，可以设置它们间的换算关系，系统会自动换算。

三、 产品供应和仓储信息

- **Customer Lead Time:** 客户提前期，从确认客户订单的时点，到能给客户发货的时点的估算时间。通常是一个平均时间。以天为单位。
- **Manufacturing Lead Time:** 生产提前期，从本产品的生产订单下单的时点，到产品产出的时间。通常是一个平均时间。以天为单位。

- **Warranty (months):** 产品保证期，以月为单位。
- **Storage Localisation:** 产品在仓库中的存放位置。如行、列、货架。
- **Procurement Location:** 采购库位，用于复式库存管理。产品采购引起的库存移动的源库位，通常是虚拟库位。
- **Production Location:** 生产库位，用于复式库存管理。产成品入库引起的库存移动的源库位，可以是车间，也可以是虚拟库位。
- **Inventory Location:** 盘点库位，用于复式库存管理。产品盘盈或盘亏引起的库存移动的源库位（盘盈）、目标库位（盘亏）。

Information	Procurement & Locations	Prices & Suppliers	Descriptions	Packaging	Accounting
Delays ? Customer Lead Time: 7.00 ? Manufacturing Lead Time: 1.00 Warranty (months): 0.00		Storage Localisation Rack: <input type="text"/> Row: <input type="text"/> Case: <input type="text"/>			
Counter-Part Locations Properties ? Procurement Location: Procurements ? Production Location: Production ? Inventory Location: Inventory loss		Miscellaneous Active: <input checked="" type="checkbox"/> Company: <input type="text"/>			

四、 产品价格和供应商信息

- **Cost Price:** 产品标准成本价（采购价）。
- **Costing Method:** 成本计算方法，有 Standard Price（标准价法）和 Average Price（移动平均法）两种。
- **Sale Price:** 产品标准售价。
- **Variant Price Margin:** 产品价格系数。
- **Variant Price Extra:** 产品价格附加。如果没有设置价格表，则产品的销售价格计算方法是：实际售价 = Sale Price * Variant Price Margin + Variant Price Extra。如果使用价格表，则实际售价根据价格表的规则计算，此两字段无用。

Information	Procurement & Locations	Prices & Suppliers	Descriptions	Packaging	Accounting
Base Prices ? Cost Price: 1.00 ? Sale Price: 1.00 Variant Price Margin: 1.00 Variant Price Extra: 0.00 ? Costing Method: Standard Price					
Supplier Information					
<< First < Previous [0 - 0 of 0] Next > Last >>					
Seq	Partner	Delivery Delay	Minimal Quantity	Partner	Product Code
Empty row for data					

- **Supplier Information:** 供应商信息（订货提前期、最低采购量、适用价格表等）。当根据补货要求（如库存量低于最小库存规则设定的数量），系统自动生成采购单时，必须要设置好产品的供应商信息，否则会发生自动补货异常错误。

 **Supplier Information**

? Partner: 维佳科技

Priority: 1

? Partner Product Name:

? Partner Product Code:

? Delivery Delay: 1

? Minimal Quantity: 0.00

Pricelist

<< First < Previous [0 - 0 of 0] Next > Last >>

Quantity	Unit Price
10	100

- **Partner:** 供应商对应的业务伙伴。
- **Priority:** 当该产品有多个供应商时，Priority 小的优先考虑。
- **Partner Product Name:** 产品名和采购名（供应商产品名）不同时，此处填入采购用产品名。
- **Partner Product Code:** 供应商产品编码。
- **Delivery Delay:** 从订单确认到仓库收到货所需的天数，自动采购时该时间用于计算下单提前期。
- **Minimal Quantity:** 该供应商所能接收的最低采购批量。
- **Quantity 和 Unit Price:** 根据采购批量设定的采购价格。

五、 产品描述和包装

- **产品描述:** 有 Description、Sale Description、Purchase Description 三个。后两者会出现在于销售、采购订单的备注里。
- **产品包装:** 一种产品可以有多种包装形式，如单件、小盒、大盒、箱装等。可以指定每种包装方式的产品件数、条形码、体积、皮重、毛重、包装材料等。

Information Procurement & Locations Prices & Suppliers Descriptions Packaging Accounting

Packaging

<< First < Previous [0 - 0 of 0] Next > Last >>

Sequence	EAN	Quantity by Package	Type of Package
----------	-----	---------------------	-----------------

Import | Export << First < Previous [0 - 0 of 0] Next > Last >>

六、 产品财务信息

- **Income Account:** 产品收入科目，产品销售业务的会计分录的贷方，通常是“6001 主营业务收入”。借方是应收账款。
- **Expense Account:** 产品成本科目，产品采购业务的会计分录的借方。对于商业流通企业，通常是“1042 在途物资”。贷方是应付账款。
- **Stock Output Account:** 产品出库科目，产品出库时会计分录的借方，贷方是库存管理中库位设置时指定的科目。对于商业流通企业，通常是“6015 主营业务成本”，贷方是“1036 库存商品”。
- **Stock Input Account:** 产品入库科目，产品入库时会计分录的贷方，借方是库存管理中库位设置时指定的科目。对于商业流通企业，通常是“1042 在途物资”，借方是“1036 库存商品”。
- **Sale Taxes:** 产品销售时的税种。销售订单默认采用此处的税种计算税额。税种在财务模块中设置。
- **Purchase Taxes:** 产品采购时的税种。采购订单默认采用此处的税种计算税额。税种在财务模块中设置。

InformationProcurement & LocationsPrices & SuppliersDescriptionsPackagingAccounting

Sales Properties

? Income Account:

? Stock Output Account:

Purchase Properties

? Expense Account:

? Stock Input Account:

Sale Taxes

+

-

<< First < Previous [1 - 1 of 1] Next > Last >>

<input type="checkbox"/> Tax Name	Tax Included in Price	Tax Code
<input type="checkbox"/> 应交增值税(销项税)	No	

Import | << First < Previous [1 - 1 of 1] Next > Last >>Export

Purchase Taxes

+

-

<< First < Previous [1 - 1 of 1] Next > Last >>

<input type="checkbox"/> Tax Name	Tax Included in Price	Tax Code
<input type="checkbox"/> 应交增值税(进项税)	No	

Import | << First < Previous [1 - 1 of 1] Next > Last >>Export

第三节 财务数据设置

要使用 OpenERP 处理财务业务，需要预先设置如下一些基础数据：

- **会计年度 (Fiscal Years):** 在中国，一个会计年度从 1 月 1 日到 12 月 31 日。一个会计年度包含若干会计分期 (Periods)。在中国，一个会计分期是一个自然月，即从每月 1 号到月末。
- **会计科目表 (Chart Of Account):** 除了财政部统一制定的会计制度上规定的总分类科目，还包括企业自己制定的明细科目。这些科目必须预先导入到系统中。
- **分类账 (Journal):** 这是 OpenERP 用于区分不同业务类型的会计分录的概念。例如，销售相关的分录都保存在销售分类账 (Sale Journal) 中。可以自己定义 Journal 用以给分录分类。

- **税种 (Tax):** 如增值税、消费税、教育附加等不同税种。OpenERP 允许定义税种，税种中定义税额的计算方法，定义税种对应的财务科目。
- **付款条件 (Payment Terms):** 如订单生效日 30 日后所在月份的月末付款，OpenERP 支持诸如这样的付款条件设置。此处设置的付款条件可以设置到产品和订单上，这样，到了订单的付款日，系统会自动提醒出纳注意收款。
- **应税设定 (Fiscal Positions):** 定义规则，将某一税种替换成别的税种。当在给业务伙伴指定应税设定后，针对该业务伙伴的订单计税时会先按应税设定的规则替换税种后，再计算税额。

下文依次介绍上述基础数据的设置方法。

一、 会计年度设置

安装财务模块 (account module) 时，在财务模块配置步骤中，有一步是 Create a Fiscal Year。如下画面所示，此画面创建会计年度。特别注意 Charts of Account 选择 “None”，因为每个单位的会计科目都不尽相同，所以最好以后再设置自己单位的会计科目，此处暂不设。

二、 会计科目表设置

设置会计科目之前，先要设置科目类型 (Account Type)。OpenERP 的科目类型，顾名思义，就是会计科目的类别。科目类型设置菜单是：Configuration → Financial Accounting → Financial Accounts → Account Types，下图是科目类型定义画面。

- **Acc.Type Name:** 科目类型名，任意文字。
- **Code:** 科目类型编码，任意文字。
- **Sequence:** 科目类型显示顺序号，没什么特别意思。
- **Sign on Report:** 本类型会计科目在报表上的显示符号（正或负）。
- **Deferral Method:** 递延方法。本类型会计科目从本期结转至下期时的结转方法。有 None（不结转）、

Balance（余额结转）、**Detail**（原样抄至下期）、**Unreconciled**（未核销）。系统会根据设置的结转方法，自动将科目结转至下期。如果选择 **None**，则要自己手工填写结转分录，结转科目。**Unreconciled** 主要是指应税账款、应付账款，未核销的应收、应付账款科目应该结转至下期。

- **Partner account:** 没什么特别意义，仅用于信息显示。

OpenERP 的科目类型设置，除了递延方法，其它的都主要是用于信息显示，对系统处理没什么特别意义。可以按自己喜好的方式设置科目类型，如按会计六要素，即资产、负债、所有者权益、收入、费用、利润设置六个科目类型，也可以按别的标准设置。

会计科目设置。会计科目设置菜单是：**Configuration → Financial Accounting → Financial Accounts → List of Accounts**。

- **Parent:** 上级科目。
- **Account Type:** 科目类型，即前文所述中设置的科目类型。
- **Internal Type:** 科目内部类型，有 **Receivable**（应收）、**Payable**（应付）、**View**（视图）、**Consolidation**（联合）、**Others**（其它）、**Closed**（已关闭）几个类型。应收账款、应付账款通常是 **Receivable**、**Payable** 类型。**View** 类型通常用于上级科目，即有下级科目的科目其类型设为 **View**。**View** 类型的科目不可用于记账，只可用于汇总。
- **Reconcile:** 可核销，通常只有应收、应付科目才勾选。勾选了的科目，系统允许通过转帐或付现进行核销。
- **Display History:** 勾选的话，系统打印总帐时，会打印科目的明细记录，而不仅仅只打印科目汇总值。
- **Default Taxes:** 附加于科目上的税种。
- **Consolidated Children:** 和 **Consolidation**（联合）类型使用。不是很清楚其用法。

Account

SearchFormCalendarGanttGraphProcess

SaveSave & EditCancel

<< First < Previous [5/14] Next > Last >>

Name: 上海福达公司Code: 101401

Parent: 1014 应收账款Company: 北京鹏瑞公司Account Type: 收入

General InformationNotes

? Secondary Currency: ? Outgoing Currencies Rate: At Date

? Reconcile: Active:

? Display History: Internal Type: Receivable

Default Taxes:

<< First < Previous [0 - 0 of 0] Next > Last >>

<input type="checkbox"/> Tax Name	Tax Included in Price	Tax Code
-----------------------------------	-----------------------	----------

Import | Export << First < Previous [0 - 0 of 0] Next > Last >>

Consolidated Children:

<< First < Previous [0 - 0 of 0] Next > Last >>

<input type="checkbox"/> Code	Name	Debit	Credit	Balance	Company	Currency
-------------------------------	------	-------	--------	---------	---------	----------

Import | Export << First < Previous [0 - 0 of 0] Next > Last >>

REPORTS

Analytic Cl

General Le

Account Ba

LINKS

Unreconcil

Entries

三、 分类账（Journal）设置

分类账设置菜单是：Configuration → Financial Accounting → Financial Journals

- Type:** 分类账类型，有 Sale（销售）、Purchase（采购）、Cash（现金）、Gernal（通用）、Situation（状况）五个选项。当确认销售凭证，系统自动生成销售分录时，系统会自动查找“Sale”类型的分类账，然后生成销售分录。如果系统找不到销售分类账，则此时的处理会报错。因此，系统至少要设置一个销售分类账。同理，系统至少要设置一个采购分类账（用于系统自动生成采购分录）和现金分类账（用于系统自动生成收、付款分录）。如果不使用系统的自动生成分录的功能，而是完全手工录入分录，则不需要设置上述三个分类账。
- View:** 分类账显示视图。该视图定义记账凭证录入界面上应显示的分录字段及显示顺序。可以自定义视图，参见后文“分类账视图定义画面”。
- Entry Sequence:** 该分类账的记账凭证的凭证号序列生成器。即每当录入新分录时，系统调用该序列生成器生成下一个凭证号。由于中国会计要求一个月内的所有凭证都按顺序编号，因此，为了保证同一序号系列，最好所有分类账都用同一个 Entry Sequence。Entry Sequence 在“管理员”菜单下配置。
- Invoice Sequence:** 凭证号用的序列生成器。当分类账用于给凭证（而不是分录）分类时，要求设置该

33

序列生成器。

- **Default Debit Account:** 默认借方科目。当在该分类账上输入分录时，如果已经输入了“贷方”，系统自动使用这里设置的科目生成“借方”行。本设置主要是辅助分录输入。
- **Default Credit Account:** 默认贷方科目。当在该分类账上输入分录时，如果已经输入了“借方”，系统自动使用这里设置的科目生成“贷方”行。本设置主要是辅助分录输入。
- **Entry Controls:** 设置该分类账允许使用和限制使用的会计科目。这主要用于减少分录输入错误。

Account Journal

SearchFormCalendarGanttGraphProcess

SaveSave & EditCancel

<< First < Previous [2/4] Next > Last >>

Journal Name: 销售分类帐Code: SALEActive: ☒

Type: SaleRefund Journal: ☐

General Information

Entry Controls

? View: Journal View

? Currency:

Analytic Journal:

? Entry Sequence: Account Journal

? Invoice Sequence:

Default Debit Account: 101401 上海福达公司

Default Credit Account: 101401 上海福达公司

? User: 系统管理员

? Centralised counterpart: ☐

? Group invoice lines: ☐

Allow Cancelling Entries: ☐

? Skip 'Draft' State for Created Entries: ☐

分类账视图定义画面：

Journal View

Journal View: Journal View

Journal Column

<< First < Previous [1 - 5 of 14] Next > Last >>

	Sequence	Column Name	Required	Readonly	
Columns:	1	Date	Yes	No	X
	2	N. Piece	No	No	X
	3	Ref	No	No	X
	4	Partner	No	No	X
	5	Account	Yes	No	X

Import | Export

<< First < Previous [1 - 5 of 14] Next > Last >>

CloseSave

四、 税种设置

税种设置菜单：Financial Management --> Configuration --> Financial Accounting --> Taxes --> Taxes。

Account Tax Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [2/2] Next > Last >>

? Tax Name: 应交增值税(销项税) Company: 北京鹏瑞公司 Tax Code:
 Active: ☒ ? Tax Group: VAT ? Tax Type: Percent

Tax Definition Tax Declaration Special Computation

? Applicable Type: True ? Amount: 0.1700
 ? Include in base amount: ☐ ? Domain:
 Invoice Tax Account: 201501 应交增值税(销项税) Keep empty to use the income account
 Refund Tax Account: Keep empty to use the expense account
 ? Tax on Children: ☐ ? Sequence: 1
 ? Tax Included in Price: ☐ Tax Application: Sale

Account Tax

<< First < Previous [0 - 0 of 0] Next > Last >>

Tax Name	Tax Included in Price	Tax Code

Child Tax Accounts:

- **Tax Name** : a unique name required for this tax (such as 12% Sales VAT),
- **Company** : a required link to a company associated with the tax, such as the Main Company,
- **Tax Group** : VAT or Other , used to determine which taxes on products can be substituted by taxes on partners,
- **Tax Type** : a required field directing how to calculate the tax: Percent , ``Fixed`` , None or Python Code , (the latter is found in the Compute Code field in the Special Computation tab),
- **Applicable Type** : a required field that indicates whether the base amount should be used unchanged (when the value is True) or whether it should be processed by Python Code in the Applicable Code field in the Special Computation tab when the value is Code),
- **Amount** : a required field whose meaning depends on the Tax Type, being a multiplier on the base amount when the Tax Type is Percent , and a fixed amount added to the base amount when the Tax Type is Fixed ,
- **Include in base amount** : when checked, the tax is added to the base amount and not shown separately,
- **Domain** : is only used in special developments, not in the core Open ERP system,
- **Invoice Tax Account** : a General Account used to record invoiced tax amounts, which may be the same for several taxes or split so that one tax is allocated to one account,
- **Refund Tax Account** : a General Account used to record invoiced tax refunds, which may be the same as the Invoice Tax Account or, in some tax jurisdictions, must be separated,
- **Tax on Children** : when checked, the tax calculation is applied to the output from other tax calculations specified in the Childs Tax Account field (so you can have taxes on taxes), otherwise the calculation is applied to the base amount on the transaction,
- **Tax included in Price** : when checked, the total value shown includes this tax,
- **Tax Application** : selects whether the tax is applicable to Sale, Purchase or All transactions,
- **Child Tax Accounts** : other tax accounts that can be used to supply the figure for taxation. You can use child taxes when you have a complex tax situation that you want to hide your end users from. For example, you might define a motor mileage expenses product with a composite tax made up of two child taxes – a non-reclaimable private element and a reclaimable business element (which is the case in some European

countries). When your staff come to claim motor mileage, they do not need to know about this taxation, but the accounting impact of their claim will be automatically managed in Open ERP.

The fields above apply the taxes that you specify and record them in the general accounts but don't provide you with the documentation that your tax authorities might need. For this use the Tax Declaration tab to define which Tax Cases should be used for this tax:

- **Invoices/Base Code** : tax case to record the invoiced amount that the tax is based on,
- **Invoices/Tax Code** : tax case to record the invoiced tax amount
- **Credit Notes/Refund Base Code** : tax case to record the refund invoice amount that the tax is based on,
- **Credit Notes/Refund Tax Code** : tax case to record the refund invoice tax amount.

第四节 库存信息设置

一、 理解 OpenERP 的库存管理

首先，OpenERP 是通过记录产品在库存间的移动来记录库存数量的。OpenERP 的数据库中没有库存数量的字段，它的库存数量是实时计算得来的。OE 的库存管理不记录库存数量，它记录的是库存移动，记录在 stock_move 表中。例如采购进货时，入库单上的每一条进货明细对应 stock_move 的一条记录，记录该货物移动的源库位、目标库位，数量，移动日期，移动状态（已发生 or 尚未发生）。计算产品库存时，汇总该产品的 stock_move，入库减去出库的 Move，即得该产品的总库存。如果只要计算某库位的产品库存，汇总目标库位是该库位的 move 数量，减去源库位是该库位的数量，即得库存。或许有人要问，采购入库时，是从供应商处提取货物，记入我方仓库，库存移动的源库位在哪？是的，OpenERP 的库存管理中，为了解决此类问题，设置了若干虚拟库位，分别对应 Supplier、Customer、Production 等。采购入库时，从供应商的虚拟库位 Supplier Location 移动到本公司库位，销售出库时，从本公司库位移动到客户的虚拟库位 Customer Location。生产原料出库时，从公司库位到虚拟库位 Production Location，车间成品入库时，从 Production Location 移动到公司库位。甚至库存盘盈盘亏时，系统记录的是公司库位到虚拟库位 Inventory loss Location 的库存移动。

在业务伙伴的 Stock Properties 中，可以为每一业务伙伴设置 Supplier Location 和 Customer Location。Supplier Location、Customer Location 可以是真实的库位，也可以是为了记录库存移动而虚设的库位。在产品的 Counter-Part Locations Properties 中，可以为每一产品设置 Production Location 和 Inventory loss Location。同样的，Production Location、Inventory loss Location 可以是真实的，也可以是虚设的。例如，可以将每个生产车间当成一个生产库位进行设置。

其次，关于虚拟库存。虚拟库存在 OpenERP 中叫 virtual stock, futurestock，顾名思义，是将来库存，不是现在的库存（现在的库存叫 real stock）。虚拟库存的概念是，将要进货的数量减去将要出货的数量，加上现存的数量，即得虚拟库存。因为 2 天内和 10 天内将要进货、出货的数量不同，所以虚拟库存的数量根据计算日期而异。虚拟库存的意义是，例如，某笔销售业务，客户要在 5 天后提货 100 件。此时，销售员要检查库存是否足够，但他应该检查的是 5 天后的库存，而不是现在的库存。所以，对实际业务而言，重要的是虚拟库存，而不是实际库存。

第三，Warehouse 和 Location。Warehouse 是仓库，对应公司真实存在的仓库。一个仓库通常包含若干个库位（Location），如，合格品区、收货区、辅助区、不合格品区。在 OpenERP 中，Warehouse 包括收货区、出货区、存货区三个库位。当然，简单情况，可以将三个区设成同一个库位。Location 是库位，库位可以真实存在，也可以是虚拟的。库位下面还可以包括若干子库位，如下图所示。库存计算、库存移动都是针对库位的，不是针对仓库的。所以，库位是 OpenERP 库存管理的核心概念。

<input type="checkbox"/>	Location Name
<input type="checkbox"/>	Physical Locations
<input type="checkbox"/>	Physical Locations/鹏瑞实业有限公司
<input type="checkbox"/>	Physical Locations/鹏瑞实业有限公司/Output
<input type="checkbox"/>	Physical Locations/鹏瑞实业有限公司/主库区
<input type="checkbox"/>	Physical Locations/鹏瑞实业有限公司/直营店库区
<input type="checkbox"/>	Physical Locations/鹏瑞实业有限公司/次品库区
<input type="checkbox"/>	Partner Locations

第四，库位链。OpenERP 中，可以设置库位链来跟踪货物的移动顺序。例如，假如某仓库的入库程序是，收货区 → 验货区 → 存货库区。那么，可以设置库位链：收货区库位 → 验货区库位 → 存货库区库位，当收货人确认后，系统自动增加收货区到验货区的库存移动，验货人确认后，系统自动增加验货区到存货库区的库存移动。也即是，设置好库位链后，货物可以手动或者自动在库位链间移动。

二、 库位和仓库设置

库位设置，菜单 Stock Management → Configuration → Locations，画面如下。

Stock location

Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel

<< First < Previous [4/13] Next > Last >>

Location Name: 主库区

Active: ☒

Location Type: Internal Location

Inventory Account: 1036 库存商品

Parent Location: 鹏瑞实业有限公司

Location Address:

Icon:

Chained Locations

Chained Location Type: Customer

Chained Location If Fixed:

Automatic Move: Manual Oper

Chained Delay (days): 0

Localisation

Corridor (X): 0

Shelves (Y): 0

Height (Z): 0

Additional Information

REPORTS

Location Cor

Lots by locat

Location Ove

ACTIONS

Products by

LINKS

Stock by Lot

Products

- Location Type: 库位类别，可选类别有：(supplier,Supplier Location)，(view,View)，(internal,Internal Location)，(customer,Customer Location)，(inventory,Inventory)，(procurement,Procurement)，(production,Production)。Supplier Location、Customer Location 分别对应采购入库的源库位、销售出库的目标库位，可用于业务伙伴的 Stock Properties 的设置。View 通常是虚拟库位，用于囊括子库位。Internal Location 通常是真实的存放货物的库位，和我们一般理解的库区的概念最接近。Inventory 是盘盈盘亏的库位，库存盘点时，盘盈的话，相当于从 Inventory Location 移动到公司库位，盘亏的话相当

于从公司库位移动到 Inventory Location。Production 通常对应到生产车间的在产库存。Procurement 用于产品设置中的 Counter-Part Locations Properties 的 Procurement 库位的设置，具体含义尚不太清楚。

- **Inventory Account:** 入库、出库时的会计科目。产品出库时会计分录的贷方，产品入库时会计分录的借方。对于商业流通企业，通常是“1036 库存商品”。
- **Parent Location:** 父库位。
- **Location Address:** 库位地址，真实库位的话，可以设置库位地址。
- **Icon:** 库位显示用图标，没啥含义。
- **Chained Location Type:** 库位链类型，可选项有：(none,None), (customer,Customer), (fixed,Fixed Location)。None 表示没有库位链，Fixed 表示本库位链接的下一库位是 Chained Location If Fixed 所设置的库位，Customer 表示本库位链接的下一库位是出入库单上的业务伙伴的“Customer Location”。
- **Chained Location If Fixed:** 本库位的库位链的下一库位。
- **Automatic Move:** 是否自动移动到库位链的下一库位，可选值有：('auto', 'Automatic Move'), ('manual', 'Manual Operation'), ('transparent', 'Automatic No Step Added')。Automatic Move，系统自动创建一条本库位到下一库位的库存移动记录。Manual Operation，要求手工确认后，系统才会创建库位链间的库存移动记录。Automatic No Step Added，自动沿库位链移动到最终库位，不生成中间各库位间移动的库存移动记录。
- **Chained Delay (days):** 移动到下一库位时，延迟天数。
- **Localisation:** 库位所处地理位置坐标。可以用于计算本库位到客户所在地的距离，从而为客户选择最佳出货仓库。

仓库设置，菜单 Stock Management → Configuration → Warehouse，画面如下。

三、 设置最小库存规则

最小库存规则，也叫订货点，设置菜单 Stock Management → Automatic Procurements → Minimum Stock Rules，画面如下。

- **Name:** 订货点（OP，Order Point）名称。
- **Location:** 本订货点适用库位。
- **Product:** 本订货点适用产品。比较麻烦的是，OpenERP 只能逐个产品设置订货点，不能批量设置。
- **Min Quantity:** 当库存低于此数量时，启动自动订货流程。系统每天检查库存，如果低于此值，系统自动生成采购申请。
- **Max Quantity:** 自动补货数量。当库存低于最小库存时，系统自动补货到此值。
- **Qty Multiple:** 自动补货时，采购数量调整到此值的倍数。例如，根据 Max Quantity 计算得到补货数量为 13，Qty Multiple 的值为 5，则系统会自动调整补货数量为 15（Qty Multiple 的倍数）。
- **Purchase Order:** 当自动生成了采购订单时，此处记录当前补货的订单号。

Minimum Stock Rules		Search	Form	Calendar	Gantt	Graph	Process	Print	Help
Save Save & Edit Cancel		<< First < Previous [-/0] Next > Last >>							
Name:	OP/00001		Active:		<input checked="" type="checkbox"/>				
Warehouse:			Location:						
Product:			Product UOM:						
? Min Quantity:			? Max Quantity:						
? Qty Multiple:	1		Purchase Order:						
[CUSTOMISE]									

四、 库存移动 (stock_move)

菜单 Stock Management → Traceability → Low Level → Stock Moves 可以查看库存移动记录。每条库存移动记录表示一个产品从源库位到目标库位的一次移动。

Stock Moves		Search	Form	Calendar	Gantt	Graph	Process	Print	Help
Save Save & Edit Cancel		<< First < Previous [1/2] Next > Last >>							
Move Information									
Source Location:	Suppliers		Dest. Location:		主库区				
Product:	[prtA] A产品		Product UOM:		PCE				
Quantity:	2.00		Name:		PO:[prtA] A产品				
Date Created:	04/30/2010 13:		? Date:		04/30/2010 13:				
Priority:	Urgent		Dest. Address:						
Packing List:	PACK8		? Production Lot:						
? Tracking Lot:									
Split in production lots									
Move State									
Status:	Available		Cancel		Make Parcel				

REPORTS
 Print Item Label
ACTIONS
 Upstream traceability
 Downstream traceability
LINKS
 Related Picking

- **Source Location:** 本库存移动的源库位。
- **Dest. Location:** 本库存移动的目标库位。
- **Product:** 本库存移动的产品。
- **Date:** 本库存移动计划的移动时间，如果移动已经完成，记录实际移动时间。
- **Packing List:** 本库存移动所属的装箱单。
- **Production Lot:** 生产批号，通常是入库或出库时填写，记录本库存移动的产品生产批号。
- **Tracking Lot:** 跟踪批号，通常是入库或出库时填写，跟踪本库存移动的产品追溯批号。

- **Split in production lots:** 如果本次库存移动的产品含有多个批次，则需要根据批次裂开成多条移动记录。
- **Status:** 可能的状态有, ('draft', 'Draft'), ('waiting', 'Waiting'), ('confirmed', 'Confirmed'), ('assigned', 'Available'), ('done', 'Done'), ('cancel', 'Canceled')。

五、 装箱单 (Packing List)

在 OpenERP 中，出库单、入库单、库存调拨单，内部实现都是用 Packing List 对象，对应的数据表是 stock_picking。一个 Packing List 包含若干条库存移动记录 (stock_move)。菜单 Stock Management → Traceability → Low Level → Packing

Packing list

Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [6/10] Next > Last >>

Partner: 维佳科技公司 Shipping Type: Getting Goods Reference: PACK8

Date Order: 04/30/2010 13:09: Planned Date: 04/30/2010 13:09

General Information Notes Others info

Moves

Name	Product	Quantity	UOM	Packing List	Production Lot	Source Location	Dest. Location	Date	Status
PO: [prtA] A产品	[prtA] A产品	2.00	PCE	PACK8		Suppliers	主库区	04/30/2010 13:09:16	Available

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

Split move lines in two

Status: Available Validate Cancel

REPORTS

Packing list

ACTIONS

Create invoice

Return packing

- **Partner:** 本装箱单的业务伙伴，如果是入库单，通常是供应商，如果是出库单，通常是客户。
- **Shipping Type:** 可能的值有, ('out', 'Sending Goods'), ('in', 'Getting Goods'), ('internal', 'Internal'), ('delivery', 'Delivery')。对应着出库单、入库单、库间调拨单、发运单。
- **Reference:** 装箱单名称 (编号)。
- **Date Order:** 装箱单生成时间。
- **Planned Date:** 计划时间，即库存移动时间。
- **Moves:** 库存移动明细。
- **Split move lines in two:** 将本装箱单裂开成多张装箱单。
- **Status:** 可能的状态有, ('draft', 'Draft'), ('auto', 'Waiting'), ('confirmed', 'Confirmed'), ('assigned', 'Available'), ('done', 'Done'), ('cancel', 'Canceled')。
- **Validate:** 确认装箱单，这会使得装箱单的状态由 'Available' 变成 'Done'。
- **Create invoice:** 画面右边的 Actions，用于生成本装箱单对应的财务票据。
- **Return packing:** 画面右边的 Actions，用于退货，当本装箱单上的物品不合格，需要退货时，适用此功能。

六、 库存盘点

菜单 Stock Management → Periodical Inventory 用于库存盘点。OpenERP 的库存盘点，内部实现是，每一个被盘点的产品，都生成一条库存移动记录。

Lot Inventory Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [4/4] Next > Last >>

Inventory: 盘点测试001 Date create: 05/01/2010 21:58:58

General Informations Posted Inventory

Stock Inventory Lines

<< First < Previous [1 - 2 of 2] Next > Last >>

Location	Product	Quantity	Product UOM
主库区	[prtA] A产品	0.00	PCE
主库区	[prtB] B产品	0.00	PCE

Import | Export << First < Previous [1 - 2 of 2] Next > Last >>

Status: Draft Confirm Inventory

ACTIONS
Fill Inventory
Set Stock to 0

Stock Inventory Lines: 盘点单明细。

Fill Inventory: 右边的 Action 功能，将选定库位的产品及库存数量，全部列出显示到盘点单上。

Set Stock to 0: 右边的 Action 功能，和 Fill Inventory 功能一样，不同的是，库存数量全清零。

Confirm Inventory: 使得盘点结果生效，这会影响库存数量，并将自动生成盘盈盘亏的会计分录。

第四章 进销存案例实施

北京鹏瑞实业有限公司是一家商业公司，他们从北京维佳科技公司购进 A 产品，然后卖给上海福达公司。本章将基于 OpenERP 搭建一个进销存系统，支持北京鹏瑞实业有限公司处理下述业务：

- **产品采购**，从询价开始，到采购物品入库。
- **产品销售**，从报价开始，到销售产品出库。
- **库存管理**，采购、销售中的入库、出库处理，以及库存查看、盘点等业务。
- **采购、销售业务记账**，支持产品采购、销售业务的财务自动记账（系统自动生成相关会计分录）。
- **收付款处理**，支持采购付款、销售收款及自动记账（系统自动生成收付款会计分录）。

搭建支持上述业务的进销存系统，首先要安装相关模块，而后要设置一些基础数据。支持上述业务，需要设置下述基础数据：

- **财务科目**
- **分类账 (Journal)**，包括销售、采购、现金分类账
- **税种**，本案例将在系统中设置增值税
- **业务伙伴**，本案例的业务伙伴有北京维佳科技公司和上海福达公司。
- **产品**，本案例仅有一个产品，即 A 产品。

第一节 模块安装

本案例需要安装下述模块：

- 产品管理 (the product module),
- 库存管理 (the stock module),
- 财会管理 (the account module),
- 采购管理 (the purchase module),
- 销售管理 (the sale module).

本案例将先安装 account module，然后安装 sale module，其它模块作为这两个模块的依赖模块，会自动安装。

第一步，创建一个新数据库账套。如下图所示，特别注意**不要载入演示数据**。



超级管理员密码:	<input type="password"/>
新建帐套名称:	<input type="text" value="casel"/>
载入演示数据:	<input type="checkbox"/>
默认语言:	<input type="text" value="Chinese (CN) / 简体中文"/>
管理员密码:	<input type="password"/>
确认密码:	<input type="password"/>
<input type="button" value="确定"/>	

第二步，登录账套，安装 Minimal Profile（安装中，配置公司名称为“北京鹏瑞实业有限公司”）。

选择一个配置

配置: Minimal Profile

配置文件规定了预选模块的具体需要.这些配置已经安装以帮助你发现系统的不同的配置.这是一个配置里面含300多个模块.
你以后可以通过管理员菜单安装更多的模块.

取消

下一个

第三步，安装 account 模块。

点击菜单：Administration → Modules Management → Modules → Uninstalled modules，选择模块 account，点击进入。而后点击右下方的按钮“Schedule for Installation”，然后点击右边工具条上的“Apply Scheduled Upgrades”安装 account 模块。安装时，先点击画面右下方的“Start Upgrade”按钮开始安装，安装完后，点击右下方的按钮“Start configuration”开始配置。配置过程中，其它步骤都按缺省配置，唯有下图的“创建会计年度”时，Chart of Account 要选择 None。

Create a Fiscal Year

? Name: 2010会计年度

? Code: 2010

Start Date: 01/01/2010

End Date: 12/31/2010

Periods: Month

Select Chart of Accounts

Charts of Account: None

Skip

Continue

第四步，安装 sale 模块。

和第三步同样的方法，安装 Sale 模块。在 sale 模块的配置步骤中，下图所示的“配置销售订单物流方式”中，Shipping Default Policy 选择“Invoice Based on Deliveries”。这个选项告诉系统根据发货单开发票。

Configure Sale Order Logistic

Packing Default Policy: Direct Delivery

Shipping Default Policy: Invoice Based on Deliveries

? Steps To Deliver a Sale Order: Delivery Order Only

Cancel

Set Default

[CUSTOMISE]

第五步，设置用户语言为英文。

由于目前的界面汉化很多术语不太对，本案例采用英文界面讲述。如果前述的安装步骤中，界面显示一直是中文，则点击画面右上角的“首选项”，设置用户语言为英文。如果已经是英文，则不必修改。

首选项

? 密码:

注意更换密码后需重登录才生效.

语言:

English

 时区:

Asia/Chongqing

签名:

系统管理员

取消

保存

第六步，确认。
上述模块装好以后，首页显示的菜单如下图所示：

Menu

Toolbar

Partners

Products

Financial Management

Human Resources

Stock Management

Purchase Management

Manufacturing

Sales Management

Administration

Menu

Configuration

Partners

Partner Addresses

第二节 财务数据设置

本案例需要设置的财务数据有：会计科目表、分类账（Journal）、和税种（增值税）。

一、 设置会计科目表

第一步，设置科目类型。
点击菜单：Financial Management → Configuration → Financial Accounting → Financial Accounts → Account Types 。新建下表中的科目类型。设置方法及表中各字段的含义，请参看第三章。

Acc. Type Name	Code	Deferral Method	Partner account	Sequence	Sign on Reports
视图	view	none	FALSE	5	Positive
收入	income	unreconciled	FALSE	5	Positive
支出	expense	unreconciled	FALSE	5	Positive
现金	cash	balance	FALSE	5	Positive
其它	other	none	FALSE	5	Positive

第二步，设置会计科目表。

点击菜单：Financial Management → Configuration → Financial Accounting → Financial Accounts → List of Accounts。新建下表中的会计科目。设置方法及表中各字段的含义，请参看第三章。未列入表中的字段，按系统自动提供的缺省值设置。

Code	Name	Account Type	Parent	Internal Type	Reconcile
0	迷你科目表	视图		view	不勾
1001	库存现金	现金	0	other	不勾
1002	银行存款	现金	0	other	不勾
1036	库存商品	其它	0	other	不勾
1014	应收账款	收入	0	view	不勾
101401	上海福达公司	收入	1014	receivable	勾选
101402	维佳科技公司	收入	1014	receivable	勾选
2011	应付账款	支出	0	view	不勾
201101	维佳科技公司	支出	2011	payable	勾选
201102	上海福达公司	支出	2011	payable	勾选
2015	应交税费	其它	0	view	不勾
201501	应交增值税(销项税)	其它	2015	other	不勾
201502	应交增值税(进项税)	其它	2015	other	不勾
6001	主营业务收入	其它	0	other	不勾

第三步，确认科目表设置。

会计科目设置好以后，点击菜单：Financial Management → Configuration → Financial Accounting → Financial Accounts → Chart of Accounts，应看到下图所示的会计科目表。

Chart of Accounts		
Toolbar	Code	Name
迷你科目表	1001	库存现金
	1002	银行存款
	▼ 1014	应收账款
	101401	上海福达公司
	101402	维佳科技公司
	1036	库存商品
	▼ 2011	应付账款
	201101	维佳科技公司
	201102	上海福达公司
	▼ 2015	应交税费
	201501	应交增值税(销项税)
	201502	应交增值税(进项税)
	6001	主营业务收入

二、 设置分类账（Journal）和税种

第一步，设置分类账。

点击菜单：Financial Management → Configuration → Financial Accounting → Financial Journals ，新建下述三个分类账。设置方法及表中各字段的含义，请参看第三章。未列入表中的字段，按系统自动提供的缺省值设置。

Journal Name	Code	Type	View	Entry Sequence	Default Debit Account	Default Credit Account
采购分类账	PUJ	Purchase	Journal View	Account Journal		
销售分类账	SAL	Sale	Journal View	Account Journal		
现金分类账	CASH	Cash	Cash Journal View	Account Journal	银行存款	银行存款

第二步，设置增值税。

点击菜单：Financial Management → Configuration → Financial Accounting → Taxes → Taxes ，新建下表两个税种。设置方法及表中各字段的含义，请参看第三章。未列入表中的字段，按系统自动提供的缺省值设置。

Tax Name	Amount	Invoice Tax Account	Tax Application
应交增值税(销项税)	0.17	201501 应交增值税(销项税)	Sale
应交增值税(进项税)	0.17	201502 应交增值税(进项税)	Purchase

第三节 业务伙伴和产品信息设置

第一步，点击菜单：Partners → Partners，新建下表两个业务伙伴。设置方法及表中各字段的含义，请参看第三章。未列入表中的字段，按系统自动提供的缺省值设置。

Name	Customer	Supplier	Contact Name	Address Type	Street	Account Receivable	Account Payable
上海福达公司	勾选	不勾	上海福达公司	Default	五一南路 68 号	101401 上海福达公司	201102 上海福达公司
维佳科技公司	不勾	勾选	维佳科技公司	Default	海淀区夏阳村	101402 维佳科技公司	201101 维佳科技公司

第二步，设置产品分类。

点击菜单：Products → Configuration → Products Categories，新建一个产品分类“所有产品”。

第三步，设置 A 产品信息。

点击菜单：Products → Products，新建一个 A 产品。各字段值如下（未说明者按系统默认值）：

- Name: A 产品
- Code: 001
- Category: 所有产品
- Income Account: 6001 主营业务收入
- Expense Account: 1036 库存商品
- Sale Taxes: 应交增值税(销项税)
- Purchase Taxes: 应交增值税(进项税)

第四节 采购业务测试

一、 采购订单

点击菜单：Purchase Management → New Purchase Order，新建下图所示订单。订单从业务伙伴“维佳科技公司”采购 A 产品 100 件。

- **添加产品明细：**新建的订单，第一次点击 Purchase Order Lines 行右边的按钮时，系统保存订单，第二次点击，则弹出产品选择窗口。在该窗口选择产品、输入数量、价格等信息，添加产品明细行。
- **金额计算：**点击右下方的 Compute 按钮，系统自动计算订单的税额和总价。税额计算方法是，根据产品上设置的税种计算。A 产品的增值税率是 17%。
- **订单确认：**新建的订单，当添加好产品明细后，点击右下方的“Confirm Purchase Order”按钮，表示确认按钮。实际工作中，这一步通常是已报主管同意后，确认该订单。而后，将订单传真给供应商确认，供应商签字确认回传后，点击右下方的“Approved by Supplier”按钮。此时，系统自动产生一张采购凭证送财务部门财务部门据此核对供应商送来的发票以及付款和做帐。一张收货单送仓管部门，仓管部门据此收货。
- **Received:** 仓库收到货物的话，此处自动打勾（订单对应的 Packing List 入库确认后，系统自动勾选本字段）。
- **Invoiced&Paid:** 出纳付完采购款项，核销(Recocile)本采购单的应付账款后，此处自动打勾。
- **Invoicing Control:** 在 Delivery & Invoices Tab 页，Invoicing Control 控制采购凭据生成策略。选项有：From Order: a draft invoice will be pre-generated based on the purchase order. The accountant will just have

to validate this invoice for control.

From Picking: a draft invoice will be pre-generated based on validated receptions.

Manual: no invoice will be pre-generated. The accountant will have to encode manually.

在版本 5.06 中，实际测试的结果和上述解说稍有出入。选择 From Order 时，采购单 “Approved by Supplier” 后，会根据采购单上的明细金额自动生成财务凭据送往财务。选择 From Picking 时，系统不会自动生成财务凭据，需要在仓库收货时候，点击画面上右边的 “Create Invoice” 创建采购凭据。此种方式，可以根据实收数量和价格手工调整采购凭据上的相应项目。选择 Manual 时，只能由财务自己做凭证，采购和仓库收货人员都不能创建采购凭据。当不使用 OE 的财务模块记账时，也许这种方式很合适。

- **Invoice:** 在 Delivery & Invoices Tab 页，如果系统生成了采购凭据，显示采购凭据的单据号，否则，显示空白。
- **订单状态(Order Status):** 采购订单的可能状态有（前者为内部状态，后者为界面上显示的状态描述）：
('draft', 'Request for Quotation'), ('wait', 'Waiting'), ('confirmed', 'Confirmed'), ('approved', 'Approved'), ('except_picking', 'Shipping Exception'), ('except_invoice', 'Invoice Exception'), ('done', 'Done'), ('cancel', 'Cancelled')
draft: 询价单
wait: 似乎系统未使用
confirmed: 主管确认了的采购订单
approved: 供应商签字确认了的采购订单（订单传真给供应商，供应商签字后传回，然后在系统中点击 “供应商通过” 按钮）
done: 已结单，采购物品已入库（对应 Packing List 已 Done），应付款项已核销（财务上已确认采购凭证，且款项已付，付款记录已冲减/核销本凭证。）
cancel: 取消了的采购单
except_picking: 由于产品不合格等原因，拒收单（以验货员拒收而异常结单的采购单）。
except_invoice: 财务拒绝付款等原因而异常结单的采购单
- **确认产品库存:** 点击菜单 Products → Products, 进入 A 产品画面，可以看到此时 A 产品的 Real Stock 没有变化，Virtual Stock 增加了 100 件。

Purchase Order Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [2/2] Next > Last >>

Order Reference: P0002 ? Date: 03/13/2010 Invoiced & Paid: ☐
 Warehouse: 鹏瑞实业 ? Origin: Received: ☐

Purchase Order Delivery & Invoices Notes

Supplier: 维佳科技公司 Address: 维佳科技公司海淀区夏阳
 ? Pricelist: Default Purchase Price Partner Ref.:

Purchase Order Lines

<< First < Previous [1 - 1 of 1] Next > Last >>

Scheduled date	Description Product	Quantity	Product UOM	Unit Price	Subtotal
03/13/2010 18:03:51	[001] A产品 [001] A产品	100.00	PCE	1.00	100.00 x

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

Untaxed Amount: 100.00 Taxes: 17.00 Total: 117.00 Compute

? Order Status: Request for Quo Confirm Purchase Cancel Purchase

二、 采购财务处理

点击菜单：Financial Management → Invoices → Supplier Invoices，可以看到系统产生了一张下图所示的采购凭证。

- **Invoice Address:** 发票地址，系统自动提取业务伙伴中设置的发票地址。
- **应收账款:** 系统自动提取业务伙伴中设置的“Account Payable”科目。本例是“201101 维佳科技公司”。
- **采购成本科目:** 系统自动提取产品中设置的“Expense Account”科目，本例是“1036 库存商品”。
- **Taxes:** 系统自动提取产品中设置的税种及税种的记账科目，本例是“应交增值税(进项税)”。
- **凭证确认:** 点击画面右下方的“Validate”按钮，确认凭证。此时，系统自动生成采购业务的会计分录。并将应付账款记入对应业务伙伴账户。
- **确认应收账款:** 进入业务伙伴“维佳科技公司”的表单，确认“Total Payable”应增加采购凭证中的金额，本例是“117.00”。

Supplier Invoice Search Form Calendar Gantt Graph Process

New Edit Duplicate Delete << First < Previous [2/2] Next > Last >>

Journal: 采购分类账 Type: Supplier Invoice Currency: 元
 Partner: 维佳科技公司 Invoice Address: 维佳科技公司海淀区夏阳村 ? Account: 201101 维佳科技公司

Invoice Other Info Payments

? Due Date: ? Bank Account:
 Free Reference P3PO2 Total: 117.00

Invoice lines

<< First < Previous [1 - 1 of 1] Next > Last >>

Product	Account	Taxes	Analytic Account	Quantity	Unit Price	Subtotal	Description
[001] A产品	1036 库存商品	(1)		100.00	1.00	100.00	[001] A产品

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

Taxes

<< First < Previous [1 - 1 of 1] Next > Last >>

Tax Description	Base	Amount
应交增值税(进项税)	100.00	17.00

Compute Taxes Untaxed: 100.00
 Tax: 17.00
 Total: 117.00
 ? Paid/Reconciled: ☐ State: Draft ? Residual: 117.00

Validate Cancel

- 确认采购业务会计分录：点击菜单 Financial Management → Entries Encoding → Entries Encoding by Line，选择“采购分类账”，打开，可看见下述画面所示的会计分录。这个分录正是：

借：1036 库存商品 100
 2015 应交税费 - 201501 应交增值税(进项税) 17
 贷：2011 应付帐款 - 201101 维佳科技公司 117

Effective date	Move Ref.	Partner Ref.	Account Name	Maturity date	Debit	Credit	Tax
03/13/2010 2	P3PO2	维佳科技公司	201101 维佳科技公司	PO002	0.00	117.00	
03/13/2010 2	P3PO2	维佳科技公司	201502 应交增值税(进项税)	应交增值税(进项税)	17.00	0.00	
03/13/2010 2	P3PO2	维佳科技公司	1036 库存商品	[001] A产品	100.00	0.00	

三、 采购付款

月底，当拿到银行对账单后，可以根据对账单上的收、付款记录批量处理收付款的会计分录。点击菜单：Financial Management → Entries Encoding → Entries by Statements → New Statement，新建一个对账单。

Statement Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [3/3] Next > Last >>

Name: St. 03/131 Date: 03/13/2010

Journal: 现金分类账 Currency: 元

Period: 03/2010 Import Invoice

Starting Balance: 900.00 Ending Balance: 783

Entry encoding Real Entries

Statement lines

<< First < Previous [1 - 1 of 1] Next > Last >>

Date	Ref.	Name	Type	Partner	Account	Amount	Reconcile
03/13/2010		支付货款	Supplier	维佳科技公司	201101 维佳科技公司	-117.00	x

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

State: Draft Balance: 783.00 Compute Confirm

- **Starting Balance:** 输入对账单上的期初额（上月余额）。
- **Ending Balance:** 输入对账单上的本月余额。
- **Statement lines:** 依次输入对账单上各收付款行。当 Type 中选择“Supplier”、Partner 选择“维佳科技公司”时，系统自动调出“维佳科技公司”的应付款，填入 Amount 栏。应付款显示为负数、应收款显示为正数。Reconcile 栏填入要核销的采购凭证，对应凭证核销以后，该采购订单上的字段“Invoiced & Paid”会被 Check。如果不填，系统不知道本笔付款对应哪张或哪几张采购凭证。这样，系统就不能自动减少对供应商的应付款记录。
- **Balance:** 点击右下方的“Compute”按钮时，系统用“Starting Balance”加/减“Statement lines”中各

收付款项后的余额。按下“Confirm”按钮确认时，“Ending Balance”和“Balance”的值必须相等，否则系统报错，不予确认。

- **对账单确认：**按下右下方的“Confirm”按钮确认后，系统自动生成付款的会计分录。分录中的科目“1002 银行存款”是对账单的 Journal “现金日记账”中设置的默认科目。系统生成的分录正是：

借: 2011 应付帐款 - 201101 维佳科技公司 117.00

贷: 1002 银行存款 117.00

Effective date	Move	Name	Statement	Partner Ref.	Account Ref.	Debit	Credit	Tax	Analytic Account	Status
03/13/2010	*4	支付货款	St. 03/131	维佳科技公司	1002 银行存款	0.00	117.00			Valid
03/13/2010	*4	支付货款	St. 03/131	维佳科技公司	201101 维佳科技公司	117.00	0.00			Valid

四、 采购入库处理

点击菜单：Stock Management → Incoming Products，打开系统生成的收货单。

Input Packing List
Search Form Calendar Gantt Graph Process

New Edit Duplicate Delete
<< First < Previous [1/1] Next > Last >>

Partner: 维佳科技公司 Origin Reference: PO001 Invoice Control: Not from Packing
Reference: PACK2 Planned Date: 03/13/2010 17:38:01 Back Order:

General Information Notes

Stock Moves
<< First < Previous [1 - 1 of 1] Next > Last >>

Product	Quantity	Product UOM	Dest. Location	Production Lot	Status
[001] A产品	100.00	PCE	Stock		Available

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

Unreceived Products

Status: Available Products Received Cancel

- **收货：**点击右下方的按钮“Products Received”表示接收货物，此时系统会提示输入接收货物的数量。如果数量低于收货单上显示的“100”（对方分批送货的情况），系统自动产生一张剩余货物数量的收

货单，供下次收货用。“Unreceived Products”表示拒收货物，这会导致采购订单的状态变为“收货异常”。

- **Origin Reference:** 引起本 Packing List 的单据引用，如果是采购入库，通常是采购单号。如果是仓管员手工新建的收货单，此处通常是手工输入的入库单号或发票号，以便于财务对账。
- **Invoice Control:** 表示本 Packing List 对应的财务凭据（或说财务发票）的状态，有 Not from Packing, to be invoice, Invoiced 三种情况。Not from Packing 表示不是基于 Packing List 开票，也就是基于采购订单开票或手工开票。to be invoice 表示待开票，这出现在基于 Packing List 开票的情况，需要点击画面右边的“Create Invoice”开具财务凭据。Invoiced 表示已开票。
- **Back Order:** 表示单据链中的前一单据。如某 Packing List，应收货 10 件，但由于种种原因，第一次只收到 6 件，则 OE 会自动将该 Packing List 分裂成两个，一个 6 件，已收货；一个 4 件，待收货。此时，后者的 Back Order 就是前一 Packing List。前者的 Back Order 为空。
- **确认库存:** 货物收下后，进入 A 产品表单画面，确认库存。此时 A 产品的 Real Stock 和 Virtual Stock 都增加了 100 件。

第五节 销售业务测试

一、 销售订单

点击菜单：Sales Management → Sales Orders → New Quotation，新建报价单，向上海福达公司销售 A 产品 50 件。

Sales order Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [1/1] Next > Last >>

Order Reference: S0001 Customer Ref: Picked: ☐
 Shop: 鹏瑞实业 Date Ordered: 03/13/201 Paid: ☐

Sale Order Other data History

Customer: 上海福达公司 ? Ordering Contact: 上海福达公司
 Invoice Address: 上海福达公司 Shipping Address: 上海福达公司
 Pricelist: Public Priceli Analytic Account:

Sales order lines

<< First < Previous [1 - 1 of 1] Next > Last >>

Description	Qty	UoM	Discount (%)	Unit Price	Net Price	Subtotal
001] A产品	50.00	PCE	0.00	1.00	1.00	50.00 x

Import | Export << First < Previous [1 - 1 of 1] Next > Last >>

Untaxed Amount: 0.00 Taxes: 0.00 Total: 0.00 Compute

? Order State: Quotation Confirm Order Cancel Order

- **订单确认**: 添加好产品明细后, 点击右下方的“Confirm Order”按钮, 确认订单。此时系统自动生成一张发货单送交仓库。根据“凭证生成方式”的设置, 系统可能自动生成一张销售凭证给财务部门, 也可能要求再按一次“Create Invoice”按钮才生成凭证(手工确认后生成凭证的方式)。本例是后者。
- **Picked**: 已发货的话, 此处自动打勾(订单对应的 Packing List 出库确认后, 系统自动勾选本字段)。
- **Paid**: 核销(Recocile)本销售单的应收账款后, 此处自动打勾。
- **订单状态(Order Status)**: 销售订单状态有,
 ('draft', 'Quotation'), ('waiting_date', 'Waiting Schedule'), ('manual', 'Manual In Progress'),
 ('progress', 'In Progress'), ('shipping_except', 'Shipping Exception'),
 ('invoice_except', 'Invoice Exception'), ('done', 'Done'), ('cancel', 'Cancelled')
 draft: 报价单
 waiting_date: 尚不太清楚, 似乎是留待指定日期再处理的单子
 manual: 主管已确认, 等待工厂生产的单子。如果订单上有“按订单生产”的产品, 会进入这个状态。
 progress: 等待仓库发货及财务收款的订单。如果订单上全是“按库存生产”的产品, 则主管确认后直接进入此状态。
 done: 已结单, 仓库已经出库, 财务已经确认凭证, 已收到款项, 且已核销应收款。
 cancel: 取消了采购单
 shipping_except: 仓库无法发货而异常结束的单子。
 invoice_except: 财务部门认为风险太大等原因而异常结束的单子。

- **确认业务伙伴事件：**打开业务伙伴“上海福达公司”的表单，在“History”页，可以看到系统自动将客户的本次购买行为记录在事件中了。
- **Packing Policy：**销售订单的“Other Data”页。有 Partial Delivery（部分送货）、Complete Delivery（一次送货）两个选项。前者表示，仓库即使缺货，只要订单上的部分货物（部分明细项或部分数量）仓库有，就允许仓管员发货。仓管员发货时，填入各明细项的实发数，系统会自动生成一张新发货单(Packing List)，包含未发部分的明细项和数量。后者表示，系统自动检查库存，只有仓库必须有订单要求的所有明细项和数量，才允许仓管员发货。不过，实际上，系统也允许仓管员“强制发货”。前者可能导致一张订单多次发货，增加运费。
- **Shipping Policy：**有 Payment before delivery、Invoice on Order After Delivery、Shipping & Manual Invoice、Invoice from the Packing 几个选项目。
 - Payment before delivery : Open ERP creates an invoice in the Draft state. Once this is confirmed and paid the delivery is activated.
 - Invoice on Order After Delivery : the delivery order is produced when the order is validated. A draft invoice is then created when the delivery has been completed.
 - Shipping & Manual Invoice : Open ERP starts the delivery from the confirmation of the order, and adds a button which you manually click when you're ready to create an invoice.
 - Invoice from the Packing : invoices are created from the packing stage.
- **Invoice on：**有 Ordered Quantities、Shipped Quantities 两个选项。前者表示基于订单开票，后者表示基于送货单开票。当一张订单分多次送货，或者实际送货数量和订单数量不同时，两个选项会不同。

Sales order Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [1/1] Next > Last >>

Order Reference: S0001 Customer Ref: Picked: ☐

Shop: 鹏瑞实业有 Date Ordered: 03/13/2010 Paid: ☐

Sale Order Other data History

Incoterm: ? Packing Policy: Partial Delivery

Salesman: 系统管理员 ? Shipping Policy: Shipping & Manual Inv

Origin: ? Invoice on: Ordered Quantities

Payment Term: Fiscal Position:

Notes

- **Incoterm：**INCOTERMS 是 International Commercial Terms 的英文缩写，《INCOTERMS 2000》是国际商会制定的关于贸易术语的国际惯例，它于 2000 年 1 月 1 日起生效。《INCOTERMS 2000》是在《INCOTERMS 1990》的基础上经过修订而产生的。《INCOTERMS 2000》在中国被习惯称为“2000 年通则”，该通则中包含了 13 种贸易术语。按照其国际代码的第一个字母的不同，这 13 种术语被分为四个组，分别称作 E 组、F 组、C 组和 D 组。具体情况见下表：

E 组（启运）

EXW Ex Works 工厂交货

F 组（主要运费未付）

FCA Free Carrier 货交承运人

FAS	Alongside Ship	装运港船边交货
FOB	Free On Board	装运港船上交货
C 组（主要运费已付）		
CFR	Cost and Freight	成本加运费
CIF	Cost Insurance and Freight	成本保险费加运费
CPT	Carriage Paid To	运费付至
CIP	Carriage and Insurance Paid To	运费保险费付至
D 组(到达)		
DAF	Delivered At Frontier	边境交货
DES	Delivered Ex Ship	目的港船上交货
DEQ	Delivered Ex Quay	目的港码头交货
DDU	Delivered Duty Unpaid	未完税交货
DDP	Delivered Duty Paid	完税后交货

二、 销售订单财务处理

点击菜单：Financial Management → Invoices → Customer Invoices，可以看到系统产生了一张下图所示的销售凭证。

Journal: 销售分类账	Type: Customer Invoice	? Invoice Number:
Partner: 上海福达公司	Invoice Address: 上海福达公司, 五一南路68号 200120	Currency: 元
? Date Invoiced:	? Force Period:	(keep empty to use the current period)

Invoice
Other Info
Payments

? Account: 101401 上海福达公司 Description: SO001
? Payment Term:

Invoice Line
☰

<< First < Previous [1 - 1 of 1] Next > Last >>

Description	Account	Quantity	Unit of Measure	Unit Price	Discount (%)	Subtotal
[001] A产品	6001 主营业务收入	50.00	PCE	1.00	0.00	50.00

Import | Export
<< First < Previous [1 - 1 of 1] Next > Last >>

Taxes
☰

<< First < Previous [1 - 1 of 1] Next > Last >>

Tax Description	Base	Amount
应交增值税(销项税)	50.00	8.50

Compute Taxes

Untaxed: 50.00
Tax: 8.50
Total: 58.50

? Paid/Reconciled: ☐ State: Draft ? Residual: 58.50

PRO-FORMA
Create
Cancel

点击右下方的“Create”按钮，系统自动生成会计分录：

借:1014 应收账款 - 101401 上海福达公司 58.50

贷: 6001 主营业务收入 50

2015 应交税费 - 201501 应交增值税(销项税) 8.5

Effective date	Move Ref.	Partner Ref.	Account Name	Maturity date	Debit	Credit	Tax
03/13/2010 3	2010001	上海福达公司	101401 上海福达公司	SO001	58.50	0.00	
03/13/2010 3	2010001	上海福达公司	201501 应交增值税(销项税)	应交增值税(销项税)	0.00	8.50	
03/13/2010 3	2010001	上海福达公司	6001 主营业务收入	[001] A产品	0.00	50.00	

三、销售出库处理

点击菜单: Stock Management → Outgoing Products, 打开系统生成的出库单。

Packing list <div style="float: right; text-align: right;"> Search Form Calendar Gantt Graph Process Print Copy Refresh Help </div>																																																																					
New Edit Duplicate Delete				<< First < Previous [1/1] Next > Last >>																																																																	
Partner: 上海福达公司				Planned Date: 03/15/2010 18:20:55		Reference: PACK4																																																															
Invoice Status: Not from Packing				Back Order:		Origin Reference: SO001																																																															
<div style="display: flex; justify-content: space-between; border-bottom: 1px solid black;"> General Information Notes Others info </div>																																																																					
<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid black;"> Moves </div> <div style="text-align: right; padding-bottom: 5px;"> << First < Previous [1 - 1 of 1] Next > Last >> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Name</th> <th style="width: 10%;">Product</th> <th style="width: 10%;">Quantity</th> <th style="width: 10%;">UOM</th> <th style="width: 10%;">Packing List</th> <th style="width: 10%;">Production Lot</th> <th style="width: 10%;">Source Location</th> <th style="width: 10%;">Dest. Location</th> <th style="width: 10%;">Date</th> <th style="width: 10%;">Status</th> </tr> </thead> <tbody> <tr> <td>[001] A产品</td> <td>[001] A 产品</td> <td>50.00</td> <td>PCE</td> <td>PACK4</td> <td></td> <td>Stock</td> <td>Customers</td> <td>03/15/2010 00:00:00</td> <td>Confirmed</td> </tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table> <div style="display: flex; justify-content: space-between; align-items: center; padding-top: 5px;"> Import Export << First < Previous [1 - 1 of 1] Next > Last >> </div> </div>										Name	Product	Quantity	UOM	Packing List	Production Lot	Source Location	Dest. Location	Date	Status	[001] A产品	[001] A 产品	50.00	PCE	PACK4		Stock	Customers	03/15/2010 00:00:00	Confirmed																																								
Name	Product	Quantity	UOM	Packing List	Production Lot	Source Location	Dest. Location	Date	Status																																																												
[001] A产品	[001] A 产品	50.00	PCE	PACK4		Stock	Customers	03/15/2010 00:00:00	Confirmed																																																												
Split in Two																																																																					
<div style="display: flex; justify-content: space-between; align-items: center;"> Status: Confirmed Check Availability Force Availability Cancel </div>																																																																					

- **Check Availability:** 检查库存是否满足出库单要求的数量。如果数量足够，则允许出库，这时，系统会提示输入实际出库量。
- **Force Availability:** 强制允许出库。默认情况下，库存不足时不允许出库。此按钮强行出库，即使数量不足（强行部分出库）。

第六节 财务报表确认

下述菜单可以打印有关财务数据。

Financial Management

Legal Statements

Generic Reports

Account Balance

General Ledger

Print Journal

Print Central Journal

Print General Journal

- **Account Balance:** 科目余额表

- **General Ledger:** 总帐, 实际上系统会连同科目的明细账一起输出(如果在科目设置画面勾选了“Display History”, 系统默认是勾选)。
- **Print Journal:** 打印出所选分类账上的所选分期的会计分录。

第五章 OpenERP 数据导出和导入

OpenERP 的数据导入导出有如下一些方法：

- 数据库整体备份和恢复。
 - 单表数据的 CSV 格式导入导出。
 - XML 格式数据导入
 - 使用 PostgreSQL 数据库本身的 COPY 命令导入导出数据。
 - 编写数据导入导出的代码，可以调用 OpenERP 的 XML-RPC 接口，也可以 SQL 语句直接访问数据库。
- 这几个方法的复杂度依次递增，本章将介绍前四种方法。

第一节 数据库备份和恢复

在登录画面上点击按钮“数据库账套”，进入数据库操作画面。点击右上角的“备份”按钮，选择要备份的数据库，输入超级管理员密码。点击“确定”按钮，即可备份选择的数据库到文件中文件名是“databasename.dump”。如下图所示：

备份帐套

创建

删除

备份

恢复

密码

帐套:

case1

密码:

确定

在数据库操作画面，点击右上角的“恢复”按钮，选择先前备份的数据库文件。输入超级管理员密码和数据库账套名，点击“确定”按钮。这样就可将备份数据完全恢复到新数据库。

恢复帐套

创建

删除

备份

恢复

密码

文件:

bin\book\RealCase.dump

浏览...

密码:

新建帐套名称:

testdb

确定

第二节 CSV 数据导出导入

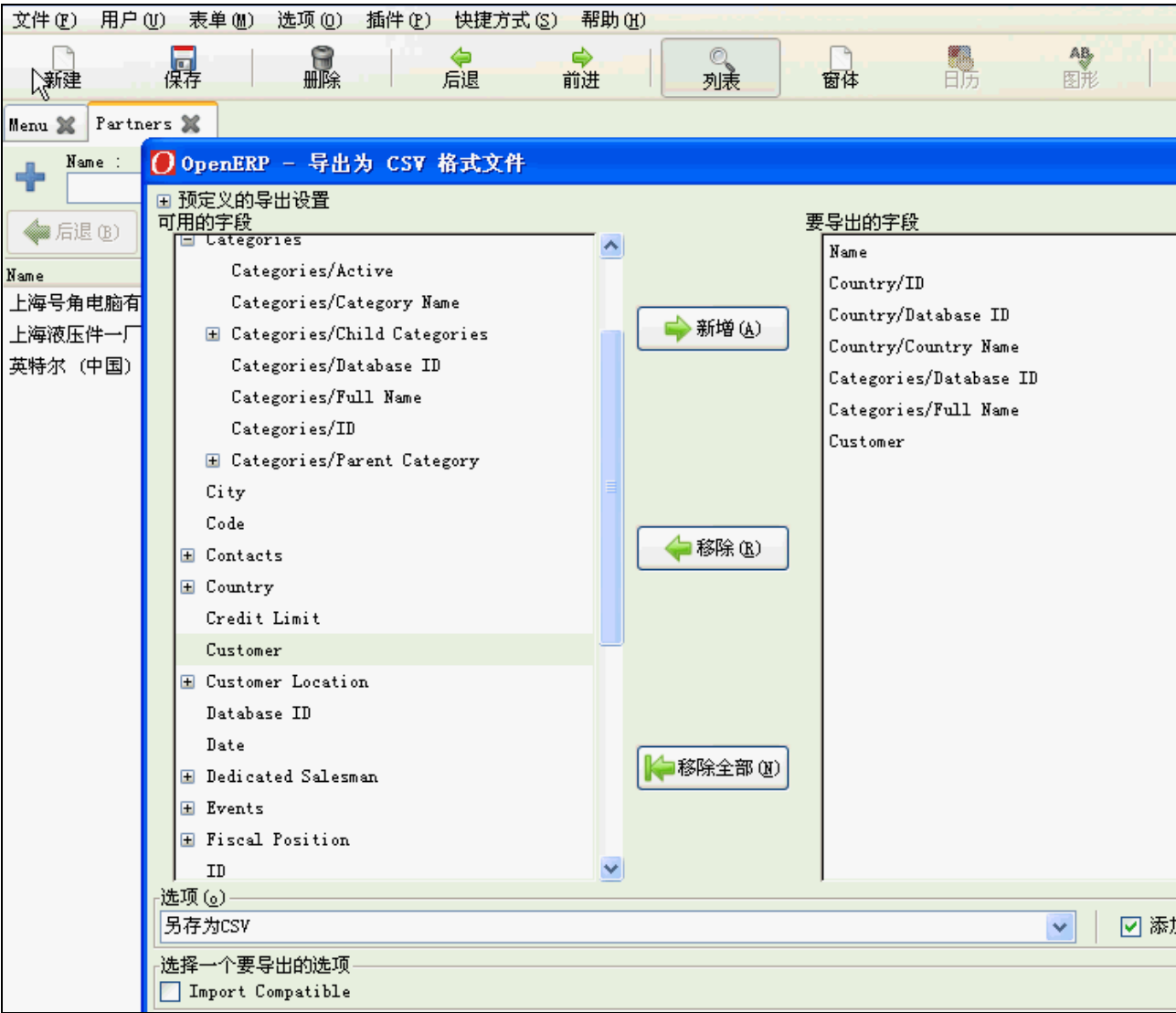
OpenERP 在任何一个列表视图上，都支持 CSV 格式的数据导出/导入。对于 Web-Client，画面左下角

有“导入|导出”按钮。GUI-Client 上，“表单”菜单的最下面，有“导入数据”、“导出数据”的菜单项。不过，写作本文时的 OpenERP 5.06 版的 Web-Client 的导出功能比较强大，但导入功能不太好用，因此，本文采用 GUI-Client 导入导出数据。

一、 数据导出

首先，为了便于理解导出数据和对象字段间的关系，也为了导出的数据能够用于导入，建议使用英文界面导入导出数据（用户语言设为英文）。本文以业务伙伴数据的导出为例说明。

菜单“Partner → Partner”，打开业务伙伴画面。而后，点击菜单“表单 --> 导出数据”，画面如下：



左边的“可用字段”栏，显示了业务伙伴对象的所有字段及其关联对象的字段。有些特别字段说明如下。

- **ID:** 安装模块或者导入数据时指定的标志该条记录的唯一 ID，这个 ID 通常在模块源码中 xxx_data.xml 文件中定义，或者 CSV 导入数据时 ID 列指定的值。模块安装时导入的数据一般都有该字段，后面手工录入的数据该字段为空。用 CSV 格式导入数据时，如果指定 ID 列，则导入的数据也有该字段。数据导入时，可以通过该字段引用先前导入的数据。例如，导入业务伙伴分类数据时 CSV 文件中指定了 ID 列，则导入业务伙伴数据时可以直接用字段名“Categories/ID”引用业务伙伴分类。数据导入时，该字段为“id”，即小写字母，否则导入会报错。
- **Database ID:** 该条记录在数据库中的 ID。所有数据都有该字段。数据导入时，可以用该字段引用数

数据库中已存在的数据。例如，导入业务伙伴时，可以用字段名“Categories/Database ID”引用数据库中的业务伙伴分类名。不过，由于查找 Database ID 不是很方便，通过 Database ID 引用数据，不如通过 ID 引用更好用。数据导入时，该字段名为“db_id”。

- **选项：**可以选择“用 Excel 打开”和“另存为 CSV”。为了便于使用该数据作为导入用，选择“另存为 CSV”。系统导出的 CSV 是 UTF-8 编码的文件，直接用 Excel 打开中文显示为乱码。解决办法是，选择用 Word 打开，然后在 Word 中另存为，选择系统默认编码，文件名不变，仍为 xxx.csv。之后再 用 Excel 打开就不是乱码了。
- **添加字段名：**打勾，这样导出的文件的第一行为字段名。带有字段名导入时比较方便。
- **Import Compatible：**导入兼容，勾选时，导出的数据能用于导入。勾选时，导出的字段名是对象内部字段名，不是界面上显示的字段名；对于下拉选择框，导出的是选择项的内部名，不是界面上显示的选择项名。不过，对于关联对象，如从业务伙伴列表画面导出业务伙伴和“业务伙伴分类 (Categories)”数据时，如果勾选本选项，则只允许导出“业务伙伴分类 (Categories)”的 ID 和 Database ID 字段，不允许导出其他字段。否则没法导入兼容。
- **导出记录数的限制，**缺省情况下，列表视图上一页最多只显示 80 条记录。导出时，5.06 版只能导出可见部分，即最多 80 条。不过，在 GUI-Client 的列表视图上，可以修改 limit 字段，改成足够大，则就可以导出所有数据。列表视图上的 limit 是每页缺省的记录条数，默认是 80。Web-Client 上，不可以修改每页的记录条数。不过 5.07 版好像没有这个导出数限制。

按图示中选择要导出的字段，导出的示例文件如下：

```
Name,Country/ID,Country/Database ID,Country/Country Name,Categories/Database ID,Categories/Full
Name,Customer
上海号角电脑有限公司,base.cn,46,中华人民共和国,,,True
上海液压件一厂有限公司,base.cn,46,中华人民共和国,,,True
英特尔 (中国) 有限公司上海分公司,base.cn,46,中华人民共和国,4,重要客户,False
,,,3,金牌供应商,
```

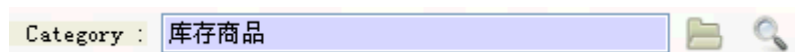
注意最后一行，该业务伙伴属于两个分类：“重要客户”和“金牌供应商”，所以导出了两行。“base.cn”是 Base 模块中定义的国家 ID（cn 为中国），“46”为“中国”在数据库中的 ID。

二、 数据导入



相比数据导出，OpenERP 的数据导入比较复杂，需要具备一些数据库知识，以及对 OpenERP 的对象间的关联有所了解，方能操作。通常，OpenERP 数据导入用的 CSV 文件的第一行写字段名。注意导入用的字段名必须是内部字段名，或者说数据库中看到的 Table 的字段名，不能用界面上看到的显示用字段名。后面各行是数据行。文件格式保存为 UTF-8 格式，CSV 可以用 Excel 编辑，而后另存为 CSV 格式，再用其它工具（很多文本工具，如 UE，EmEditor 都具备该功能）转存为 UTF-8 编码的文件。

数据导入时，由于数据表间的关联关系，导入用的 CSV 数据有所不同。数据表间的关联关系通常有三种情况，分别是多对一、多对多、一对多。


下图是多对一的情况：



下图是多对多的情况（一个产品有多个税别，一个税种适用于多个产品）：

			 添加(A)	 删除(R)
Tax Name	Tax Included in Price	Tax Code		
增值税	<input type="checkbox"/>			

下图是一对多的情况（一个业务伙伴有多个联系人）：

General	Sales & Purchases	History	Notes	Accounting
 Partner Contacts				
Contact Name : 李四				
Function : <input type="text"/>				

下面导入下述 CSV 文件的数据到业务伙伴中。

```
ID,Name,Country/ID,Categories/Database ID,Customer,Contacts/Contact Name
id,name,country:id,category_id:db_id,customer,address/name
partner_001,某某有限公司,base.cn,"3,4",True,张金玲
,,,,李四
```

- **字段名的处理**，该数据中，第一行为显示字段名，这里列出来是为了方便理解，实际导入时，要删除该行。第二行是内部字段名，该行才是导入文件中要用的字段名。可以根据导出文件的字段来编辑该行（勾选导入兼容），或者查看数据库中 Table 的字段名来编辑该行。
- **ID**，数据导入时，可以指定 ID 列（不是必须的），这样，别的关联数据导入时就可以通过 ID 引用本条数据。本例中指定本条数据的 ID 是” partner_001”。ID 可以是任意字符（”.”除外），但要保证不重复。
- **多对一关系的处理**，本例中，业务伙伴和国家是多对一关系。字段名是” country:id”（或者” country:db_id”），表示业务伙伴的 Country 字段通过国家对象的 id 字段（或者 db_id，即数据库 ID）引用国家对象。base.cn 是“中国”的 ID。当国家对象已经导入好了，且导入时导入了 ID 列，此处可以通过 ID 引用国家对象。如果没有导入 ID 列，可以用 db_id 引用国家对象。
- **多对多关系的处理**，业务伙伴和业务伙伴分类是多对多关系，处理方法和多对一关系基本相同。字段名也是”category_id:db_id”，category_id 是业务伙伴中“业务伙伴分类”的字段名，db_id 表示通过 Database ID 引用“业务伙伴分类”对象。和多对一不同的是，数据行用引号括起，填入多个 ID（本例是”3,4”），表示多对多关系。
- **一对多关系的处理**，业务伙伴和联系人间是一对多关系，即一个业务伙伴有多个联系人。一对多关系有两种处理方法，一种是按“多对一”处理，一种是按“一对多”处理。如本例，从“业务伙伴”方面看，是一对多关系，但从“联系人”方面看，是多对一关系。如果导入业务伙伴数据时，不填联系人信息；而是导入联系人对象时，填入业务伙伴信息，那么就可以按“多对一”处理。本例是按“一对多”的办法导入。一对多关系，字段名是” address/name”，表示联系人对象的 name 字段。address 是业务伙伴中联系人的字段名。导入数据格式是“多行”，即第一行含有完整数据，后面行只含有“多”方的数据。

数据导入的操作。点击菜单“表单 --> 导入数据”，选择 CSV 文件，点击“自动检测”按钮。如下图所示。



“自动检测”按钮会自动提取 CSV 文件的第一行中的字段名，如果字段名不对，它会报错，错误信息中会指出哪个字段不正确。点击确定，即可导入上述业务伙伴数据，如果数据有误，报错信息中会指明哪个数据不正确。

第三节 会计科目数据导入

本节以会计科目的导入为例，进一步补充说明 CSV 数据导入方法。OpenERP 的会计科目数据有两部分，一是科目类型，二是科目本身。导入时要先导入科目类型数据，再导入科目数据。实际工作中，建议按下述步骤导入会计科目数据。

- 1) 使用 Excel 编辑科目类型表
- 2) 编辑导入用 CSV 格式的科目类型数据表
- 3) 导入科目类型
- 4) 使用 Excel 编辑会计科目表
- 5) 编辑导入用 CSV 格式的科目数据表
- 6) 导入会计科目

下文将依照上述步骤导入进销存案例中的科目数据。注意，为了避免和已有科目代码重复而出错，进行下述实验时，最好保证账套没有安装过任何 Chart of Account 。

一、科目类型导入

- 1) 编辑科目类型表。

Acc. Type Name	Code	Deferral Method	Partner account	Sequence	Sign on Reports
视图	view	None	FALSE	5	1

收入	income	unreconciled	FALSE	5	1
支出	expense	unreconciled	FALSE	5	1
现金	cash	Balance	FALSE	5	1
其它	other	None	FALSE	5	1

表中第一行为科目类型界面上显示的字段名。必须输入的字段都要纳入数据表中。

2) 编辑 CSV 格式的科目类型表。

	Acc. Type Name	Code	Deferral Method	Partner account	Sequenc e	Sign on Report s
id	name	code	close_metho d	partner_accoun t	sequenc e	sign
myacc_type_view	视图	view	none	FALSE	5	1
myacc_type_income	收入	income	unreconcile d	FALSE	5	1
myacc_type_expens e	支出	expens e	unreconcile d	FALSE	5	1
myacc_type_cash	现金	cash	balance	FALSE	5	1
myacc_type_other	其它	other	none	FALSE	5	1

和前一个表格相比，本表格增加了 id（注意，必须是小写）列，id 列方便其它数据导入时引用本表格的数据。本表格增加了第二行，该行显示的是数据导入用的字段名，是数据对象的内部字段名，不是界面上显示的字段名。

3) 另存为 CSV 格式，导入数据。

注意，CSV 格式文件不要包含前述表格的第一行数据，且 CSV 格式文件必须保存为 utf-8 编码格式。科目类型的 CSV 格式数据如下。

```
id,name,code,close_method,partner_account,sequence,sign
myacc_type_view,视图,view,none,FALSE,5,1
myacc_type_income,收入,income,unreconciled,FALSE,5,1
myacc_type_expense,支出,expense,unreconciled,FALSE,5,1
myacc_type_cash,现金,cash,balance,FALSE,5,1
myacc_type_other,其它,other,none,FALSE,5,1
```

二、 会计科目导入

1) 编辑科目表

Code	Name	Account Type	Company	Outgoing Currencie s Rate	Internal Type	Reconcil e
0	迷你科目表	视图	鹏瑞实业	current	View	不勾
1001	库存现金	现金	鹏瑞实业	current	Others	不勾
1002	银行存款	现金	鹏瑞实业	current	Others	不勾
1036	库存商品	其他	鹏瑞实业	current	Others	不勾

1014	应收账款	收入	鹏瑞实业	current	View	不勾
101401	上海福达公司	收入	鹏瑞实业	current	Recievable	勾
101402	维佳科技公司	收入	鹏瑞实业	current	Recievable	勾
2011	应付账款	支出	鹏瑞实业	current	View	不勾
201101	维佳科技公司	支出	鹏瑞实业	current	Payable	勾
201102	上海福达公司	支出	鹏瑞实业	current	Payable	勾
2015	应交税费	其他	鹏瑞实业	current	View	不勾
201501	应交增值税(销项税)	其他	鹏瑞实业	current	Others	不勾
201502	应交增值税(进项税)	其他	鹏瑞实业	current	Others	不勾
6001	主营业务收入	其他	鹏瑞实业	current	Others	不勾

2) 编辑 CSV 类型的科目表

	Code	Name	Account Type		Company	Outgoing Currencies Rate	Internal Type	Reconcile
id	code	name	user_type:id	parent_id:id	company_id:db_id	currency_mode	type	reconcile
myacc_0	0	迷你科目表	myacc_type_view		1	current	view	FALSE
myacc_1001	1001	库存现金	myacc_type_cash	myacc_0	1	current	Other	FALSE
myacc_1002	1002	银行存款	myacc_type_cash	myacc_0	1	current	Other	FALSE
myacc_1036	1036	库存商品	myacc_type_other	myacc_0	1	current	Other	FALSE
myacc_1014	1014	应收账款	myacc_type_income	myacc_0	1	current	View	FALSE
myacc_101401	101401	上海福达公司	myacc_type_income	myacc_1014	1	current	receivable	TRUE
myacc_101402	101402	维佳科技公司	myacc_type_income	myacc_1014	1	current	receivable	TRUE
myacc_2011	2011	应付账款	myacc_type_expense	myacc_0	1	current	view	FALSE
myacc_201101	201101	维佳科技公司	myacc_type_expense	myacc_2011	1	current	payable	TRUE

myacc_2011_02	201102	上海福达公司	myacc_type_expense	myacc_2011	1	current	payable	TRUE
myacc_2015	2015	应交税费	myacc_type_other	myacc_0	1	current	view	FALSE
myacc_2015_01	201501	应交增值税(销项税)	myacc_type_other	myacc_2015	1	current	other	FALSE
myacc_2015_02	201502	应交增值税(进项税)	myacc_type_other	myacc_2015	1	current	Other	FALSE
myacc_6001	6001	主营业务收入	myacc_type_other	myacc_0	1	current	other	FALSE

上表数据说明如下。

- 数据导入时，第一行的字段名尽量使用对象的内部字段名，即数据库中对象对应的数据表的字段名，不要使用界面显示的字段名。使用界面显示的字段名，数据导入时，系统的自动检测功能一般也能正确识别字段，但有时会出错。
- ID 字段对应的内部字段名是“id”，Database ID 字段对应的内部字段名是“db_id”。
- Account Type 字段的内部字段是“user_type”，当通过 ID 引用科目类型时，字段名写法是“user_type:id”。
- Company 字段的内部字段名是 company_id，当通过 Database ID 引用数据时，字段名写法是“company_id:db_id”。
- Internal Type 字段，使用的是选择项的内部名，不是界面上看到的选择项名。选择项的内部名可以从源码中该字段的定义处查得。

3) 另存为 CSV 格式文件。

```
id,code,name,user_type:id,parent_id:id,company_id:db_id,currency_mode,type,reconcile
myacc_0,0,迷你科目表,myacc_type_view,,1,current,view,FALSE
myacc_1001,1001,库存现金,myacc_type_cash,myacc_0,1,current,other,FALSE
myacc_1002,1002,银行存款,myacc_type_cash,myacc_0,1,current,other,FALSE
myacc_1036,1036,库存商品,myacc_type_other,myacc_0,1,current,other,FALSE
myacc_1014,1014,应收账款,myacc_type_income,myacc_0,1,current,view,FALSE
myacc_101401,101401,上海福达公司,myacc_type_income,myacc_1014,1,current,receivable,TRUE
myacc_101402,101402,维佳科技公司,myacc_type_income,myacc_1014,1,current,receivable,TRUE
myacc_2011,2011,应付账款,myacc_type_expense,myacc_0,1,current,view,FALSE
myacc_201101,201101,维佳科技公司,myacc_type_expense,myacc_2011,1,current,payable,TRUE
myacc_201102,201102,上海福达公司,myacc_type_expense,myacc_2011,1,current,payable,TRUE
myacc_2015,2015,应交税费,myacc_type_other,myacc_0,1,current,view,FALSE
myacc_201501,201501,应交增值税(销项税),myacc_type_other,myacc_2015,1,current,other,FALSE
myacc_201502,201502,应交增值税(进项税),myacc_type_other,myacc_2015,1,current,other,FALSE
myacc_6001,6001,主营业务收入,myacc_type_other,myacc_0,1,current,other,FALSE
```


第四节 XML 数据导入导出

在 OpenERP 的各个模块中，都要导入一些基础数据和演示数据。各模块目录下的 `x_data.xml` 文件通常是本模块所必须的基础数据，`x_demo.xml` 是本模块的演示数据。本节介绍 OpenERP 的 xml 导入数据的格式。

一、XML 数据导入

```
<data noupdate="1">
<record id="user_06229" model="res.users">
  <field name="login">06229</field>
  <field name="password">cs123</field>
  <field name="name">邬培英</field>
  <field name="signature">邬培英</field>
  <field name="context_lang">zh_CN</field>
  <field name="context_tz">Asia/Chongqing</field>
  <field name="action_id" ref="cshaocai.action_cshaocai_mylingId"/>
  <field name="groups_id" eval="[(6,0,[ref('base.group_user'),ref('cshaocai.group_haocaiuser')])]" />
  <field name="menu_id" ref="base.action_menu_admin"/>
</record>
</data>
```

上例的 xml 数据将导入系统用户“邬培英”到 OpenERP 中，登录名：06229，登录密码：cs123，所属权限组：base.group_user 和 cshaocai.group_haocaiuser。

- **data 行**，`<data noupdate="1">`，一个 data 下面有若干个 record。noupdate 表示，当模块升级时是否更新本条数据。对于 demo 数据，通常设置成 `noupdate="1"`，即不更新，不指定 noupdate 的话，默认值是 `noupdate="0"`。
- **record 行**，表示数据表的一条记录，record 下有若干个 field，对应数据表的各列。model="res.users" 表示数据表 res_users，即本数据记录是插入到数据表 res_users。id="user_06229"，表示本条记录的 id，导入别的数据时通过这个 id 引用本条数据。
- **field 行**，field 表示记录的一个字段值，可用属性有：name, eval, ref, search。`<field name="login">06229</field>` 表示字段 login 的值为“06229”，name 指明字段名，字段名就是数据表的列名。本例的写法，系统会把“06229”当成字符类型导入，如果希望导入数字“06229”，则用下述写法：`<field name="login" eval="6229" />`，eval 表示计算，eval 的值是一个表达式，如“3+5”。ref 表示引用别的数据，many2one 和 one2many 的数据通常用 ref 处理。ref="base.action_menu_admin" 表示引用 base 模块中的 id="action_menu_admin" 的 record，该条 record 通常定义在 base 模块的数据文件 base_data.xml 中。本例是 many2one 的数据，下例是 one2many 的数据：`<field name="groups_id" eval="[(6,0,[ref('base.group_user'),ref('cshaocai.group_haocaiuser')])]" />`。用户“邬培英”属于多个权限组，base.group_user 和 cshaocai.group_haocaiuser。也许直接用权限组的名称，而不是权限组的 id，引用权限组更方便。如“base.group_user”的名称是“Employee”，这种情况可以用 search 来实现。`<field name="groups_id" model="res.groups" search="['(name', '=', u'Employee')]" />` 表示，搜索数据表“res.groups” (model="res.groups") 中 name='Employee' 的记录，作为字段“groups_id”的值。

二、 XML 数据导出

如果直接在 OpenERP 的界面上输入了一些数据，或者创建，修改了一些对象、视图、Action、工作流等，怎么把它导出来，在别的环境上安装呢。模块 `base_module_record` 就带有这个功能。安装该模块后，菜单 Administration → Customization 下增加菜单项 Module Creation。



Start Recording, Stop Recording, Save Recorded Module，按下 Start Recording 后，系统记录数据库的一切变化，包括各数据表的记录增加、更新等，直到按下 Stop Recording。Save Recorded Module 输出系统记录下的数据变化，输出格式是 XML 数据记录。

Export Customizations As a Module，如果没有通过 Start Recording 记录界面上的操作，怎么导出系统数据呢？Export Customizations As a Module 可以完成这个功能。Export Customizations As a Module 要求选择一些对象（或者说数据表），指定一个时间，系统将导出选定的数据表中的，指定时间后的发生过变化的数据。导出结果自动生成 OpenERP 的模块形式。导出的模块主要部分就是 XML 的数据文件，期格式和前一节说的 XML 数据一样。

第五节 PostgreSQL 数据库数据导出导入

基于 CSV 文件的数据导入导出有个限制，如果某对象，没有对应的“列表”视图，则无法导出导入数据。这种情况下，可以考虑两个办法，一是自己编写代码导出导入，不过，还有一个更简单的办法是，利用 PostgreSQL 自身的命令，直接从数据库导出导入。PostgreSQL 提供了 COPY 命令，可以将数据表的数据导出到文件，以及将文件的数据导入到数据表。COPY 命令语法如下：

```
COPY tablename [ ( column [, ...] ) ]
FROM { 'filename' | STDIN }
[ [ WITH ]
    [ BINARY ]
    [ OIDS ]
    [ DELIMITER [ AS ] 'delimiter' ]
    [ NULL [ AS ] 'null string' ] ]
```

```
COPY tablename [ ( column [, ...] ) ]
TO { 'filename' | STDOUT }
[ [ WITH ]
    [ BINARY ]
    [ OIDS ]
    [ DELIMITER [ AS ] 'delimiter' ]
    [ NULL [ AS ] 'null string' ] ]
```

下面介绍使用 COPY 命令导出导入数据的具体操作步骤。如果是 Windows, 先要安装 psql 程序 (Linux 的话, PostgreSQL 安装包自带该程序)。从 sourceforge (<http://psql.sf.net/>) 下载开源软件“PostgreSQL Frontend”, 直接安装即可。

下文以“期初库存盘点”为例, 介绍怎么用 COPY 命令导入期初产品库存数量。OpenERP 中 Stock Management → Periodical Inventory 用于盘点各产品的库存量。当第一次使用 OpenERP 时, 希望能批量导入各产品的期初库存量。但是, OpenERP 的盘点单中, 各产品的库存明细在对象“Stock Inventory Lines”中(数据表 stock_inventory_line), 从 GUI Client 上无法直接导出导入该对象(该对象没有对应的列表视图), 从 Web Client 上, 虽然有“Import/Export”按钮, 但在本文写作的 5.06 版中, Import 似乎不好用。

- **新建一个草稿状的库存盘点单**, 菜单 Stock Management → Periodical Inventory → New Periodical Inventory, 该盘点单中增加一个产品的盘点明细项保存(目的是用于导出数据, 供编辑导入用数据参考)。
- **进入数据库**, 进入 PostgreSQL Frontend 的命令界面, 执行命令 psql dbname dbuser:

```
D:\work>psql case1 openpg
```

```
Password:
```

```
Welcome to psql 7.4.6, the PostgreSQL interactive terminal.
```

case1 是本文的示例数据库, openpg 是 OpenERP 默认安装的数据库用户名。输入 password, OpenERP 默认数据库密码是 openpgpwd。

- **导出库存盘点明细**, 命令如下:

```
case1=# copy stock_inventory_line to 'd://work//a11.csv' delimiter as ',';
```

```
COPY 1
```

```
case1=#
```

上述命令将数据表“stock_inventory_line”的数据导出到文件 d://work//a11.csv, 以逗号为分割符。要注意的是, 1) 文件名必须是绝对路径, 不可以是相对路径; 2) COPY 命令是以 PostgreSQL 数据库服务器的用户名执行, 因此, 必须确保该用户名拥有对指定文件的读写权限。Windows 上, PostgreSQL 数据库服务器的用户名默认是“.\openpgsvc”。默认情况下, 该用户不属于任何安全组, 为了使得该用户拥有文件读写权限, 要将该用户加入必要的安全组(如 administrators)。

本命令导出的数据如下:

```
id,create_uid,create_date,write_date,write_uid,inventory_id,location_id,product_id,product_uom,product_qty
1,1,2010-04-10 21:58:16.296,\N,\N,3,11,3,1,5
```

第一行是手工加上去的, 实际导出数据只有第二行。inventory_id 是盘点单的 Database ID, location_id 是库位, product_id 是产品, product_uom 是产品单位, product_qty 是库存数量。对应到画面显示的数据如下。

Stock Inventory Lines			
<< First < Previous [1 - 1 of 1] Next > Last >>			
Location	Product	Quantity	Product UOM
Stock	[prtC] 产品C	5.00	PCE

- **编辑盘点明细单**, 在 Excel 中编辑下述格式的 CSV 文件, 将各产品的库存数量添加到 inventory_id=3 的盘点单上。

```
inventory_id,location_id,product_id,product_uom,product_qty,create_uid,create_date
```

```
3,11,3,1,10,1,2010-04-10
```

实际操作中, 可以使用下图中的 Fill Inventory 功能, 先把所有产品都显示到该盘点单上, 再用“Set Stock to 0”将所有产品的库存设为 0。而后, 再导出该盘点单的 stock_inventory_line。这样, 所有产品及产品单位数据都有了, 只需填入盘点数量。这样可以较大的减少编辑盘点单的时间。

MAIN MENU

SHORTCUTS

Modules

Lot Inventory

Search

Form

Calendar

Gantt

Graph

Process

New

Edit

Duplicate

Delete

<< First < Previous [3/3] Next > Last >>

Inventory: 0409产品C盘点

Date create: 04/09/2010 20:50:51

ACTIONS

Fill Inventory

Set Stock to 0

- 导入盘点明细数据，编辑好盘点明细单的 CSV 数据后，删除第一行的字段行，文字转为 UTF-8 编码格式，而后，用下述命令导入到数据库中。

```
case1=#                                copy                                stock_inventory_line
(inventory_id,location_id,product_id,product_uom,product_qty,write_uid,create_date) from 'd://work//a11.csv'
delimiter as ',';
```

数据导入到数据库后，画面上确认该盘点单，则各产品的期初库存量就初始化好了。

OpenERP 应用和开发基础 (0.2 版)

作者：老肖

msn: saiwai76@hotmail.com

mail: xiangfu.xiao@gmail.com

QQ: 1417063315

更新日期：

2010 年 1 月，完成 0.1 版，即第六至第十章内容

2010 年 3 月，完成 0.2 版，即第一至第五章内容

2010 年 4 月，全面修订后全文发布

目录

第一章	OpenERP 介绍	1
第一节	OpenERP 的历史	1
第二节	OpenERP 的架构	2
一、	OpenERP 应用系统架构	2
二、	OpenERP 应用服务器架构	3
三、	PostgreSQL 数据库简介	3
第三节	OpenERP 功能和应用简介	4
一、	OpenERP 功能模块	4
二、	OpenERP 的用户群	7
第四节	OpenERP 开发简介	7
一、	数据访问功能开发	7
二、	工作流和报表开发	9
第五节	为什么选择 OpenERP	12
第二章	进销存业务概述	14
第一节	采购业务概述	14
第二节	销售业务概述	15
第三节	采购、销售的财务处理	16
第四节	仓库管理业务概述	18
第三章	OpenERP 基础业务设置	19
第一节	业务伙伴信息设置	19
一、	理解业务伙伴	19
二、	业务伙伴联系信息	20
三、	业务伙伴购销信息	21
四、	业务伙伴事件	22
五、	业务伙伴财务信息	22
第二节	产品信息设置	23
一、	产品模块配置	23
二、	产品一般信息	26
三、	产品供应和仓储信息	27
四、	产品价格和供应商信息	28
五、	产品描述和包装	29
六、	产品财务信息	30
第三节	财务数据设置	30
一、	会计年度设置	31
二、	会计科目表设置	31
三、	分类账 (Journal) 设置	33
四、	税种设置	34
第四节	库存信息设置	36
一、	理解 OpenERP 的库存管理	36
二、	库位和仓库设置	37
三、	设置最小库存规则	38
四、	库存移动 (stock_move)	39
五、	装箱单 (Packing List)	40
六、	库存盘点	41

第四章	进销存案例实施	42
第一节	模块安装	42
第二节	财务数据设置	44
一、	设置会计科目表	44
二、	设置分类账 (Journal) 和税种	46
第三节	业务伙伴和产品信息设置	47
第四节	采购业务测试	47
一、	采购订单	47
二、	采购财务处理	49
三、	采购付款	51
四、	采购入库处理	52
第五节	销售业务测试	53
一、	销售订单	53
二、	销售订单财务处理	56
三、	销售出库处理	58
第六节	财务报表确认	58
第五章	OpenERP 数据导出和导入	60
第一节	数据库备份和恢复	60
第二节	CSV 数据导出导入	60
一、	数据导出	61
二、	数据导入	62
第三节	会计科目数据导入	64
一、	科目类型导入	64
二、	会计科目导入	65
第四节	XML 数据导入导出	68
一、	XML 数据导入	68
二、	XML 数据导出	69
第五节	PostgreSQL 数据库数据导出导入	69
第六章	OpenERP 开发入门	72
第一节	基本概念	72
第二节	实例演练	74
一、	创建请假单对象	74
二、	创建视图	75
三、	创建菜单和系统动作 (Action)	77
四、	测试	79
五、	视图编辑器	80
第三节	深入数据库	81
第七章	OpenERP 的工作流开发	84
第一节	理解工作流 (Workflow)	84
第二节	实作工作流	85
一、	修改对象和视图	85
二、	创建工作流、活动和迁移	87
三、	测试工作流	91
第三节	复杂工作流实例分析	91
第四节	企业流程 (Enterprise Process)	93
第八章	OpenERP 的报表开发	96

第一节	OpenERP 报表开发的预备知识	96
一、	报表开发工具	96
二、	XSL 简介	97
第二节	OpenERP 报表简介	98
一、	基于 sxw 或 odt 模板的报表	98
二、	基于 xsl+xml 模板的报表	99
第三节	报表中文化解决方法	100
一、	让报表显示汉字	100
二、	让报表说中文	101
第四节	实作报表	101
一、	编写 RML 文件.....	101
二、	编写报表对象代码及 XML 文件.....	102
三、	测试	103
第九章	Request,Sequence and Scheduler	105
第一节	Request 和 References.....	105
一、	Request	105
二、	References 字段类型.....	106
第二节	Sequence	107
第三节	Scheduler	108
第十章	OpenERP 的权限设置	110
第一节	OpenERP 权限机制简介	110
第二节	组的权限设置方法	111
第三节	OpenERP 的预设权限组(Group).....	114
一、	预设权限组定义	114
二、	常见预设权限组	115
第四节	Domain 条件表达式	115
一、	Domain 条件表达式的应用场合.....	115
二、	Domain 条件表达式写法.....	116
第十一章	OpenERP 翻译方法	118
第一节	资源术语翻译	118
第二节	通用术语翻译	120
第三节	批量翻译或纠正翻译	121
第四节	OpenERP 国际化原理浅析	124
第十二章	OpenERP 模块开发	126
第一节	OpenERP 模块结构	126
一、	安装示例代码	126
二、	请假模块结构	127
第二节	理解 OpenERP 的代码	128
一、	对象定义	128
二、	视图、动作及菜单定义	130
三、	工作流和报表定义	131
四、	权限定义	131
第三节	服务端动作(Server Action).....	131
一、	配置 Server Action	131
二、	配置 OpenERP Mail.....	132
三、	测试通知 Mail.....	133

第六章 OpenERP 开发入门

一说到开发，大家就会想到写代码，但基于 OpenERP 的开发，不一定要写代码。OpenERP 有两种开发方式，一种不要写代码，直接在界面上操作就能完成功能开发。另外一种就是通过编写代码完成功能开发。实际上，几乎所有功能都能够通过前一种方式来完成，即无需写代码就能完成几乎所有功能。本章先讲解 OpenERP 开发的几个重要概念，其次通过界面操作开发一个简单的实例，而后进一步深入 OpenERP 的数据库揭开 OpenERP 架构的内幕。关于代码开发方法，将在以后介绍，但本章的概念对代码开发方法也非常有用，理解了本章的基本概念，代码开发就很容易上手。

第一节 基本概念

当我们操作 OpenERP 的菜单时，通常是点击菜单（如合作伙伴 → 合作伙伴），跳出对象选择画面，当选择一条记录时，跳出对象编辑画面。对象选择画面、对象编辑画面，在 OpenERP 里称为视图（View），选择画面是列表视图（Tree View），编辑画面是表单视图（Form View）。OpenERP 里的对象（Object），也叫 Model，相当于我们一般说的类（Class），对象总是对应到数据库里的数据表。例如业务伙伴对象，其对象名是“res.partner”，对应表名是“res_partner”。表里的一条记录，也就是对象的一个实例，叫资源（Resource）。

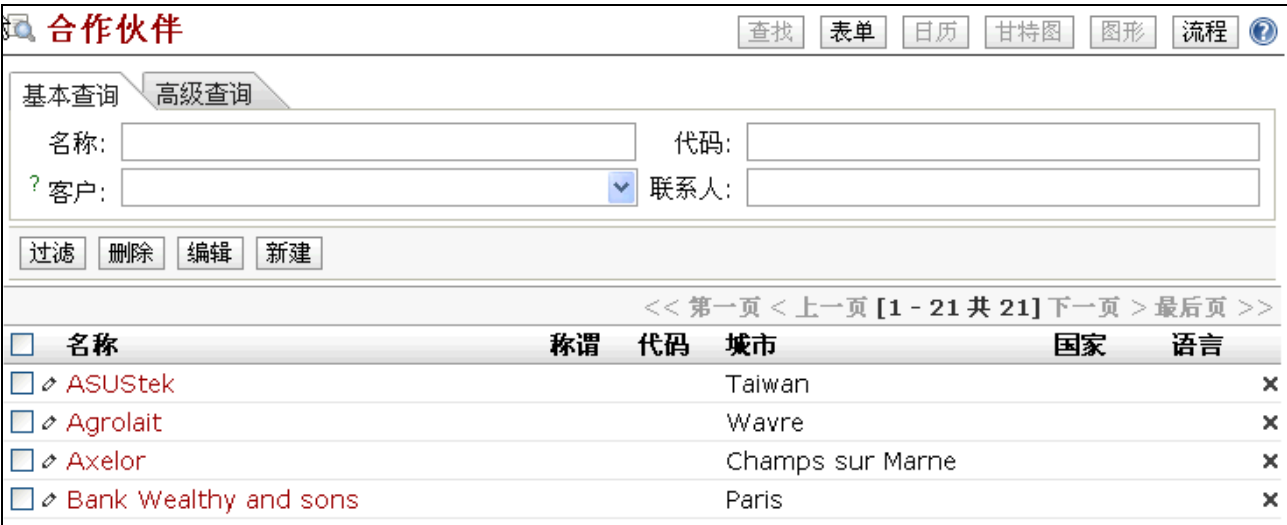


图 六-1 合作伙伴列表视图

合作伙伴

查找 表单 日历 甘特图 图形 流程

保存 保存并编辑 取消

<< 第一页 < 上一页 [1/21] 下一页 > 最后一页 >>

名称: ASUSTek 代码: ? 客户: ☒

称谓: ? 语言: ? 供应商: ☒

通用信息 购销信息 事务履历 备注

合作伙伴联系人

联系人姓名: Tang 称谓: 先生

职位: 地址类型: 默认

街道: 31 Hong Kong street 街道2:

邮编: 23410 城市: Taiwan

国家: 省市自治区:

电话: 传真:

手机: 电邮:

分类

图 六-2 合作伙伴表单视图

当点击菜单时，系统怎么知道应该跳出哪个画面，以及应该显示哪个对象的记录呢？把菜单和对应的对象、视图关联起来的是 Action。目前的汉化包中，关于 Action 的翻译有些乱，有时叫操作、有时叫动作、还有的叫功能、活动。准确的说，Action 译成中文叫“系统动作”，或简称“动作”，比较准确。当用户点击菜单时，触发 Action，Action 调用对象的 Search 方法，从数据库取得记录（资源），Action 又创建视图，显示取得的数据。

简单总结一下，OpenERP 的开发中，有如下一些重要概念：

- 对象或模型（Object or Model）：是一个 Python 的 Class，也对应到数据库的一张数据表，负责存取数据记录（Record），有 Search、Read、Write 等方法。OpenERP 在模块加载时，初始化模块中的所有对象，放入对象池。因此，数据库操作时，通常是先从对象池中取得对象，再调用对象的方法。下面分析一段典型的记录查找和读代码。

```
def _get_admin_id(self, cr):
    if self.__admin_ids.get(cr.dbname) is None:
        ir_model_data_obj = self.pool.get('ir.model.data')
        mdid = ir_model_data_obj._get_id(cr, 1, 'base', 'user_root')
        self.__admin_ids[cr.dbname] = ir_model_data_obj.read(cr, 1, [mdid], ['res_id'])[0]['res_id']
    return self.__admin_ids[cr.dbname]
```

这段代码来自文件 server\addons\base\res\res_user.py，它先从对象池取得对象'ir.model.data'，该对象负责存取数据表 ir_model_data。而后调用该方法_get_id 取得 admin 用户的 user_id，再读入 admin 对应的 user 记录的 id。

- 视图（View）：负责显示数据，最常见的视图是列表视图和表单视图。此外，还有日历、甘特图、图形、流程图等几种视图，不同的视图以不同的方式展示数据。本章主要介绍列表和表单，另外几种视图以后介绍。
- 菜单（Menu）：这个很直观，不用介绍了。
- 动作（Action）：用户操作系统时（如点击菜单、点击画面右边的工具条上的按钮等），系统的响应动作。一个 Action 包含一个对象，包含若干个视图，通常每个 Action 都包含列表和表单两个视图。当 Action 被触发时，相应的视图被调出，展示相应的对象的数据。Action 有多种类型，最常见的是

Act_Window（窗口类型），窗口类型跳出一个窗口以显示数据。此外还有 Report（报表）、Wizard（向导）等类型。本章主要介绍窗口类型。

上述概念间的关联关系，参见如下操作序列图：

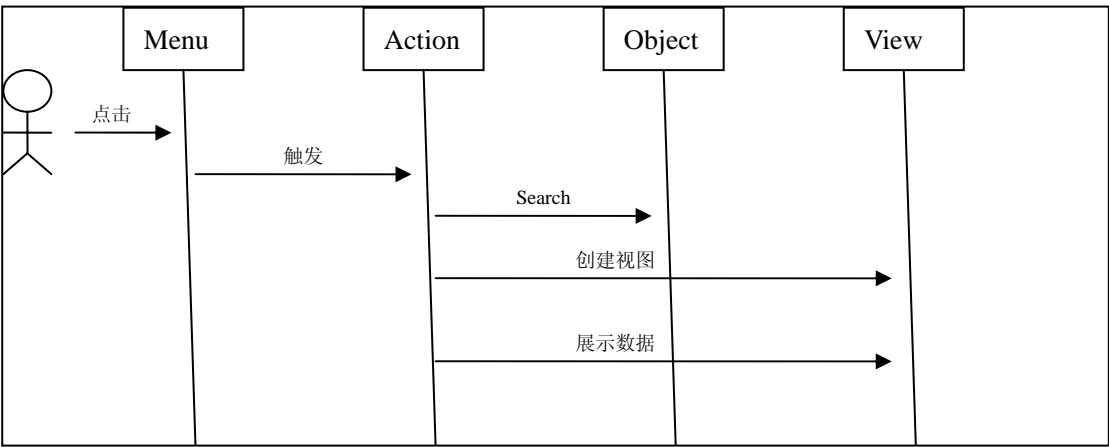


图 六-3 基本操作序列

第二节 实例演练

本节开发一个简单的请假申请功能，包括创建及编辑请假单，请假单查找。但暂不包括请假审批流程，审批流程将在后续章节开发。本节拟定的请假单包括如下信息：

- 申请人：申请人默认是当前登录用户，必填项。
- 请假天数：可以是小数，必填项。
- 开始日期：开始休假的日期，必填项。
- 请假事由：一段文本，描述请假事由，可以不填。

根据前一节的概念介绍，我们需要开发下述对象：

- 请假单对象：将请假单保存到数据库，以及从数据库查找请假单。
- 请假单视图：查找、编辑请假单的画面，包括列表和表单两个视图。
- 菜单：准备开发两级菜单，请假申请 → 请假单。点击“请假单”时，进入请假单列表视图，可以查找或创建请假单。
- 请假单动作(Action)：请假单动作把对象、视图、菜单关联到一起。

一、 创建请假单对象

点击菜单：管理员 → 自定义 → 数据库结构 → 对象，点击新建按钮，创建请假单对象。

对象名：请假单

对象：x_qingjd

字段：包括四个字段，申请人(x_shenqr)，请假天数(x_tians)，开始日期(x_kaisrq)，请假事由(x_shiyou)。

注意：从界面上创建对象时，对象及字段标识符必须是 x_开头，后面还会介绍编写代码来创建对象，编写代码创建对象时，就没有这个限制。

创建请假单对象及创建字段的参考画面如下：

主菜单

快捷菜单

Modules

模型说明

查找

表单

日历

甘特图

图形

流程

保存

保存并编辑

取消

<< 第一页 < 上一页 [1/1] 下一页 > 最后一页 >>

对象

访问权限

备注

对象名: 请假单

对象: x_qingjd

字段

字段说明

<< 第一页 < 上一页 [1 - 4 共 4] 下一页 > 最后一页 >>

名称	字段标签	字段类型	要求	仅读	可查询	手动创建	
x_shenqr	申请人	many2one	是	否	不可查询	自定义字段	×
x_tians	请假天数	float	是	否	不可查询	自定义字段	×
x_kaisrq	开始日期	date	是	否	不可查询	自定义字段	×
x_shiyou	请假事由	text	否	否	不可查询	自定义字段	×

导入 | 导出

<< 第一页 < 上一页 [1 - 4 共 4] 下一页 > 最后一页 >>

图 六-4 创建对象

字段说明

名称: x_shenqr

字段标签: 申请人

字段类型

属性

字段类型: many2one

Object Relation: res.users

Relation Field:

Field Selection: []

尺寸大小: 64

手动创建: 自定义字段

领域: []

要求: ☒

仅读: ☐

可查询: 不可查询

Translate: ☐

? 正在删除: 置空 (NULL)

图 六-5 创建字段

TODO

详细介绍字段各种类型及各项属性的含义。

二、 创建视图

点击菜单：管理员 → 自定义 → 用户界面 → 视图，点击新建按钮，依次创建请假单列表和请假单表单两个视图。

请假单列表

视图名称：请假单列表

视图类型：列表

对象：x_qingjd，即刚才创建的请假单对象，如果填写的对象标识符不正确的话，保存时会提示错误。

子栏位和继承视图不填。

优先级：1，Action 通常有多个视图，优先级表示多个视图的显示顺序，数字越小越先显示。通常列表总是最先显示，其次是表单。

视图结构：视图结构是视图中最重要的一部分，也是最复杂的部分。视图结构定义要在视图中显示哪些字段，及显示属性（如只读、必填、隐藏等）。select="1"表示，在列表视图中，可以按该字段查找资源（记录）。本视图内容如下：

```
<?xml version="1.0"?>
<tree string="请假单">
  <field name="x_shenqr" select="1"/>
  <field name="x_tians" />
  <field name="x_kaisrq" select="1"/>
  <field name="x_shiyou" />
</tree>
```

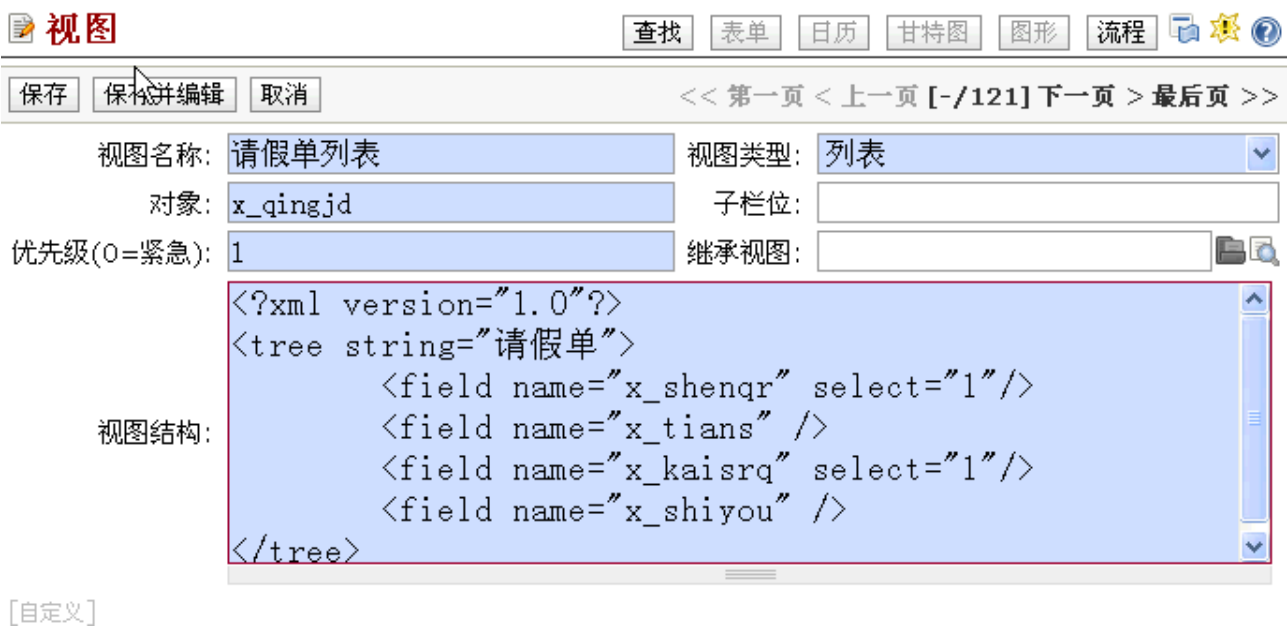


图 六-6 创建列表视图

请假单表单

视图名称：请假单表单

视图类型：表单

对象：x_qingjd

优先级：2

视图结构：本视图内容如下：

```
<?xml version="1.0"?>
<form string="请假单">
  <field name="x_shenqr"/>
  <field name="x_tians"/>
```



图 六-7 创建表单视图

三、 创建菜单和系统动作（Action）

菜单 管理员 → 自定义 → 数据库结构 → 对象 进去，查找刚才创建的“请假单”对象，点击打开，在表单的下方有个“创建菜单”的按钮，点击进去。

进去后，点击父菜单右边的查找按钮，在父菜单查找画面，新建一个“请假申请”的父菜单。

全名：不用填，系统自动生成。


菜单：请假申请，用于菜单显示的文字

序号：在菜单列表中的显示位置，数字越小越显示在上位。

父菜单：由于本菜单是顶级菜单，不要父菜单。

菜单功能：点击菜单时将触发的 Action，本菜单仅仅是父菜单，不触发 Action，故不填。

组：只有这里定义的组(group)才能看见本菜单，如果不填，表示任何组都可以看到。暂时不考虑菜单权限，故而不填。



 **菜单**

全名:



菜单:

序号:

父菜单:

菜单功能:

<< 第一页 < 上一页 [0 - 0 共 0] 下一页 > 最后一页 >>

☐ 组名

? 组:

导入 | 导出

<< 第一页 < 上一页 [0 - 0 共 0] 下一页 > 最后一页 >>

图标:






图 六-8 创建父菜单


父菜单建好后，按下图创建“请假单”菜单。点击“创建菜单”按钮，系统会自动创建菜单和菜单对应的 Action。该 Action 访问请假单对象，依次调用请假单列表和请假单表单显示数据。

菜单名:

父菜单:



对象:


 

视图

<< 第一页 < 上一页 [1 - 2 共 2] 下一页 > 最后一页 >>

序号 视图类型 视图

 1   列表 请假单列表

视图:  2 列表 请假单表单 

导入 | 导出

<< 第一页 < 上一页 [1 - 2 共 2] 下一页 > 最后一页 >>

图 六-9 创建菜单及对应动作

揭开 Action 的面纱

前面介绍过，是 Action 把对象、视图、菜单等各个元素集成到了一起，那么 Action 到底是个什么东西呢？

点击菜单：管理员 → 低级对象 → 操作 → 窗口动作，在动作对象中输入“x_qingjd”查询，得到“请假单”动作，点击打开。从 Action 编辑画面，可以看到，Action 关联了对象（x_qingjd）、视图（请假单列表和请假单表单），还有其他一些属性字段。

TODO: 解释 Action 对象各字段的含义

打开窗口

查找 表单 日历 甘特图 图形 流程

保存 保存并编辑 取消

<< 第一页 < 上一页 [1/1] 下一页 > 最后页 >>

功能名: 请假单

功能类型: ir.actions.act_window

操作对象: x_qingjd

数据源对象:

功能用法:

视图类型: 表单

视图

<< 第一页 < 上一页 [1 - 2 共 2] 下一页 > 最后页 >>

序号	视图类型	视图	
1	列表	请假单列表	×
2	列表	请假单表单	×

导入 | 导出 << 第一页 < 上一页 [1 - 2 共 2] 下一页 > 最后页 >>

过滤条件:

环境变量值: {}

? 最大显示数: 80 ? 自动刷新: 0

视图模式: tree, tree 视图参照:

图 六-10 Action 编辑画面

四、 测试

回到主菜单，画面上增加了菜单“请假申请 → 请假单”。点击请假单，进入列表视图，点击新建按钮，进入创建请假单画面。

Menu

工具条

菜单

合作伙伴

请假申请

管理员

请假单

图 六-11 新增的菜单

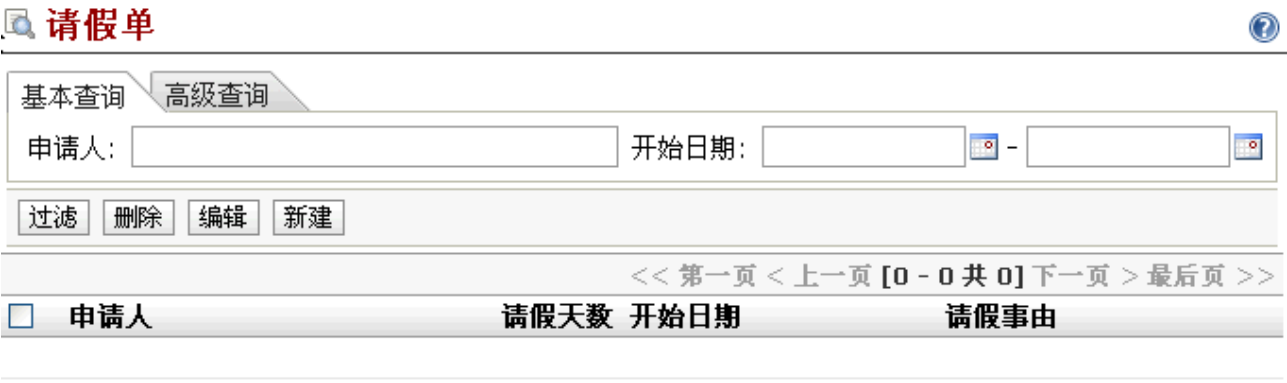


图 六-12 请假单列表视图

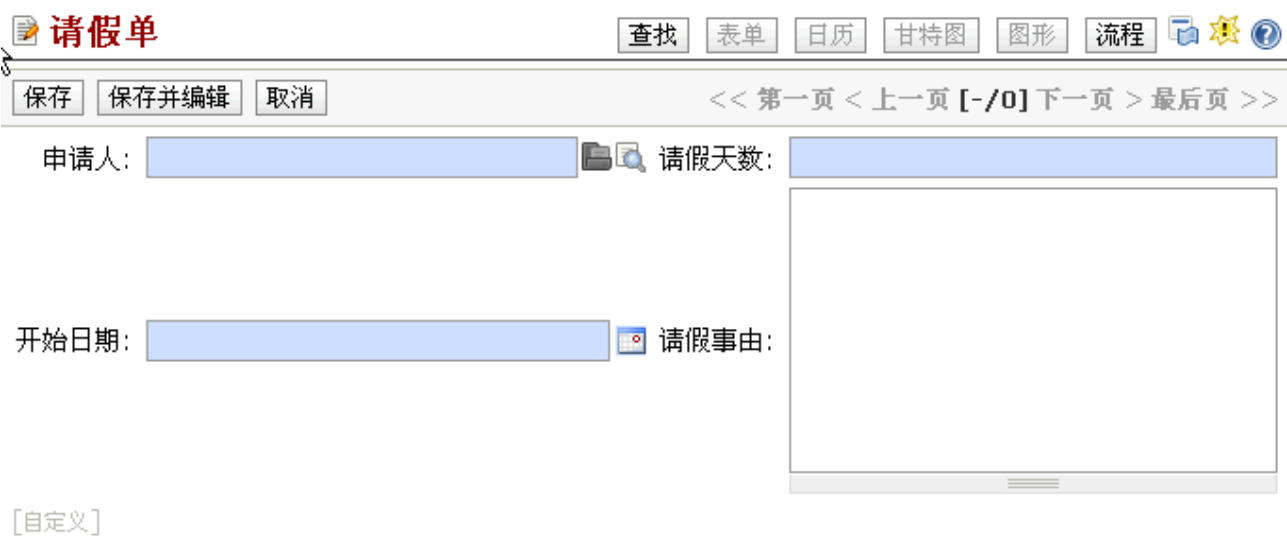


图 六-13 请假单表单视图

五、 视图编辑器

在 OpenERP 中，视图结构算是比较复杂的，不过幸运的是，除了前述的直接写 XML 编辑视图结构的方法，OpenERP 还提供了可视化的编辑方法。在请假单表单画面的左下方，有个灰色的“[自定义]”标签，鼠标移到上面，选择“管理视图”，这会列出当前 Action 所关联的所有视图。选择“请假单表单”，编辑，则弹出下述视图编辑画面。

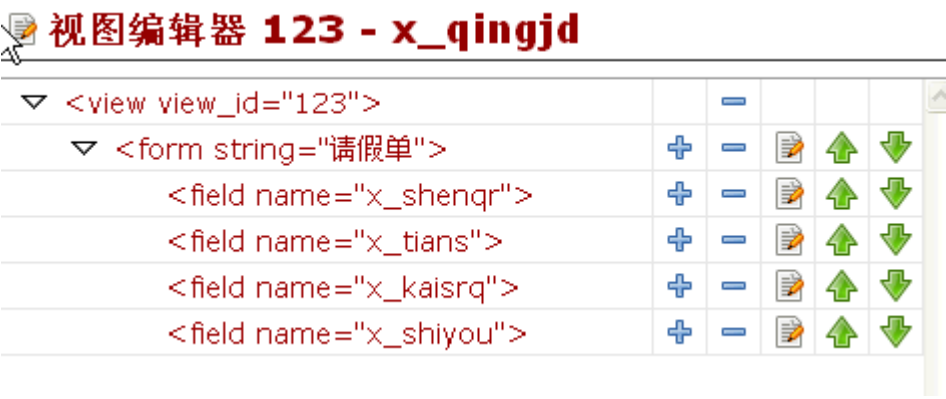


图 六-14 视图编辑画面

编辑 x_shenqr 字段，弹出下述字段编辑画面。这个画面列出了视图字段的所有可用属性。

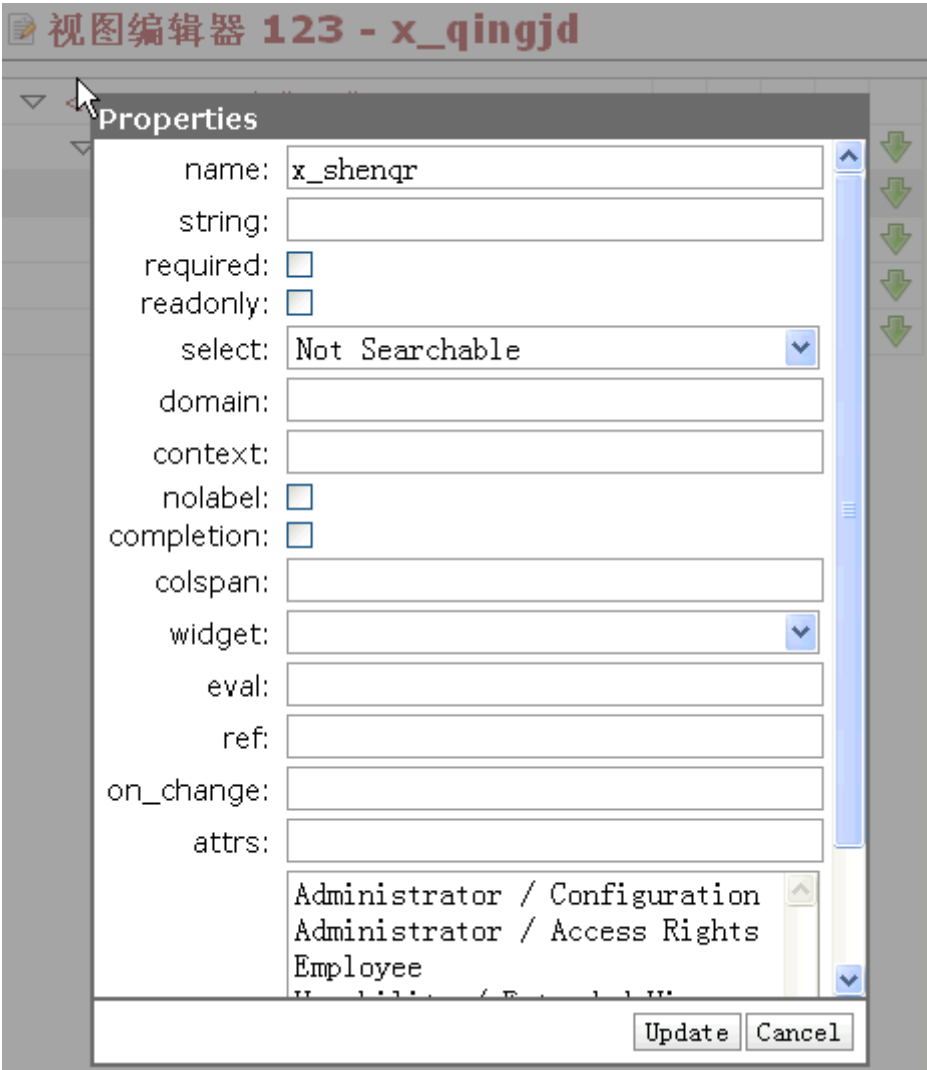


图 六-15 视图字段编辑画面

TODO

解释视图字段的每个属性及选项的含义

第三节 深入数据库

在上述界面操作中，依次创建了对象、视图、菜单和系统动作，作成了“请假申请”的功能。在界面操作的背后，OpenERP 内部做了哪些动作呢？实际开发工作中，总是会碰到这样那样的问题，只有明白了其背后动作，才知道从哪里查找原因，解决问题。本节深入 OpenERP 数据库，探求背后内幕。

对象

当在界面上创建 x_qingjd 对象时，OpenERP 在数据库中新建了一张表 x_qingjd，用于保存请假单对象，如下：

x_qingjd Table

id [PK]	create_date timestamp	write_date timestamp	write_time integer	x_shenqr integer	x_tians double	x_kaisrq date	x_shiyoud text
1	2009-12-27 2	2009-12-28 0	1	3	1.5	2009-12-29	休年假

x_qingjd 对象及其字段的信息，也写入了数据表 ir_model，字段信息在 ir_model_fields。ir_model_fields 通过 model_id 字段和 ir_model 外键关联。如下：

ir_model Table

id [PK]	model serial	name character varying(64)	state character varying(64)	info text
85	x_qingjd	请假单	manual	

ir_model_fields Table

id [PK]	model character varying(64)	model_id integer	name character varying(64)	relation character varying(64)	sequence integer	field_description character varying(255)	type character varying(64)
587	x_qingjd	85	x_shenqr	res.users	0	申请人	many2one
588	x_qingjd	85	x_tians	NULL	0	请假天数	float
589	x_qingjd	85	x_kaisrq	NULL	0	开始日期	date
590	x_qingjd	85	x_shiyoud	NULL	0	请假事由	text

视图

界面上创建的视图，写入在表 ir_ui_view，其中视图结构保存在 arch 字段中，是一段 XML 文本。如下：

ir_ui_view Table

id [PK]	name serial	model character varying(64)	type character varying(64)	arch text	field_parameters character varying(255)	priority integer
122	请假单列表	x_qingjd	tree	<?xml version="1.0"><tree default_fo		1
123	请假单表单	x_qingjd	form	<?xml version="1.0"><form version="1.0"><field name="x_shenqr" type="many2one" model="res.users" /></form>		2

菜单和动作

在界面上创建菜单（父菜单和子菜单）时，菜单信息保存在表 ir_uimenu，如下：

ir_ui_menu Table

id [PK]	parent_id integer	name character varying(64)	icon character varying(64)	create_time integer	create_date timestamp	write_time integer	write_date timestamp	sequence integer
103		请假申请	STOCK_OPEN	1	2009-12-27 2			10
104	103	请假单	STOCK_INDEX	1	2009-12-27 2			10

动作（Action）信息在表 ir_act_window，ir_act_window_view，ir_values 中。其中 Action 的基本信息在 ir_act_window 中，字段 res_model 定义了和本 Action 关联的对象。Action 和视图的关联信息在 ir_act_window_view 中，和菜单的关联信息在 ir_values 中。

ir_act_window Table

id [PK]	name serial	type character varying(32)	usage integer	view_id integer	res_model character varying(64)	view_type character varying(64)
93	请假单	ir.actions.act_window			x_qingjd	form

ir_values Table

id [PK]	name character va	key charac	key2 character varyi	model character	value text	meta text	res_id integer
84	Menuitem	action	tree_but_open	ir.ui.menu	ir.actions.act_window,93		104

Action 的最大玄机在表 `ir_values` 中，如本例，表 `ir_values` 中的字段 `model` 和 `res_id` 表示，本 Action 的触发菜单是 `ir_ui_menu` 表中的 `id=104` 的菜单项，这正是“请假单”菜单。字段 `value` 表示本 Action 触发哪个动作，本例中 `value='ir.action.act_window,93'`，表示点击菜单时触发表 `ir_action_act_window` 中的 `id=93` 的 Action，这个 Action 正是“请假单”动作。

至此，我们基本上理清了 OpenERP 的背后动作：

- 1) 从表 `ir_ui_menu` 中读取菜单信息，显示在画面上
- 2) 当用户点击菜单时，从表 `ir_values` 中找到该菜单对应的 Action
- 3) 从表 `ir_act_window` 和 `ir_act_window_view` 中，找到 Action 关联的对象和视图
- 4) 从表 `ir_model` 和 `ir_model_fields` 取得对象和字段信息，构建访问对象(本例是 `x_qingjd`)的 SQL 文
- 5) 从表 `ir_ui_view` 中取得视图信息，尤其是视图结构 XML，根据视图结构 XML 构造画面
- 6) 将对象数据显示在视图画面上。

不管是在界面上开发，还是编写代码开发，OpenERP 都是在上述表中写入对象、视图、Action、菜单信息。运行时从上述表中读取信息，访问数据库，构造画面，显示数据，处理用户动作。

第七章 OpenERP 的 workflow 开发

第一节 理解工作流 (Workflow)

考虑请假申请的流程：

- 1) 创建请假单，填入请假申请事项（创建）
- 2) 送给主管审批（送批）
- 3) 如果没有异议，主管审批通过（批准），流程结束
- 4) 主管也可能拒绝申请（拒绝），流程结束

细细分析这个流程，里面包含两个流，一个是信息流，一个是处理过程。这里的信息是请假单，请假单伴随着整个申请和审批流程。这个流程的每一步骤，都对应着一个处理动作（括号中的动作）。每个动作都对请假单做些处理，使得请假单发生变化。流程中的处理动作称为**活动 (Activity)**，请假单的变化，称为**状态变化**。当我们关注请假单状态变化时，**状态图**如下：

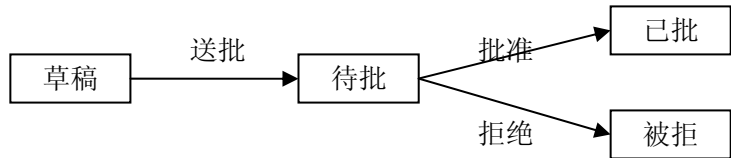


图 七-1 请假单状态图

当我们关注处理步骤时，**活动图**如下：

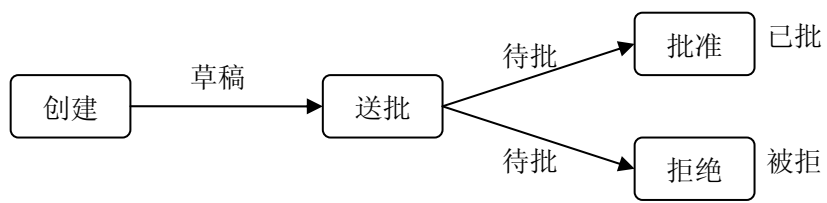


图 七-2 请假申请活动图

状态图和活动图，是工作流的两个面，前者关注信息变化，后者关注处理步骤，二者是等价的，就是说可以互相转换。

从一个状态到另一个状态，或者从一个活动到另一个活动，叫**迁移 (Transitions)**，在上述的请假申请 workflow 中，有 3 个迁移。迁移又分出**出迁移 (Outgoing transitions)**和**入迁移 (Incoming transitions)**，从活动 A 到活动 B 的迁移，对 A 来说，是出迁移，对 B 来说，是入迁移。伴随迁移，还有三个概念，一个是**触发 (Trigger)**，一个是**迁移条件 (Condition)**，还有一个是**角色 (Role)**。从 A 到 B 的迁移，不会无缘无故发生，通常有个触发事件，才会发生迁移，最典型的触发事件是用户按下某个按钮。例如，草稿状态的请假单，用户按下“送批”按钮，则迁移到“待批”状态。也有其他类型的触发事件，例如，考虑罚款单，在某个时间点之前，它是正常的罚款单，一旦到了某个时间点，变成了迟延单，迟延单的性质可能完全不一样。时间点也是一种常见的触发事件。不是每个人都有权限触发事件，例如只有主管才能执行“批准”操作，主管就是角色，也就是说，只有拥有主管角色的人才能触发“批准”迁移。触发的时候，还有个迁移条件，例如，草稿状的请假单，如果信息没填完整，则按下“送批”按钮，也不允许迁移，“信息必须完

整”就是这里的迁移条件。

在 OpenERP 中开发工作流，其实就是定义上述概念。例如本例的请假申请工作流，首先要定义待处理的对象，这里是“请假单”。其次要定义四个活动（Activity），可以按状态图定义，也可以按活动图定义。如果按前者定义，四个 Activity 是“Draft”、“Waiting Prove”、“Proved”、“Rejected”。如果按后者定义，四个 Activity 是“New”、“To Prove”、“Prove”、“Reject”。通常按状态图定义容易理解一些。再次，要定义三个迁移，以及每个迁移的触发和迁移条件。对象、活动和迁移的定义，可以在界面上操作完成，也可以编写 XML 完成。前者适合系统维护人员对系统的小修小改，后者适合新模块的开发。本章介绍界面操作的方法，XML 的方法在后续章节中介绍。

第二节 实作工作流

本节接续前一章开发的请假单对象，实作请假申请流程。根据前一节的概念介绍，我们要依次定义请假单对象，四个活动和三个迁移。请假单对象已经有了，但需要增加一个状态字段，记录请假单的状态变化。通常还要增加一个审批意见字段，为了简化问题，我们不增加该字段。

注意，由于工作流需要模块“process”支持，开始本节之前，请确保该模块已安装。如未安装，请参照本书第一章安装该模块。

一、修改对象和视图

修改对象

在请假单对象中增加 `x_state` 字段，保存请假单状态。按前一章的办法，菜单：管理员 → 自定义 → 数据库结构 → 对象，查询对象名“请假单”，打开编辑，增加字段 `x_state`。

- **名称：**`x_state`
- **标签：**状态
- **类型：**selection，selection 类型显示在界面上是下拉框，下拉框的可选值在 Field Selection 中定义。
- **Field Selection：**`[('draft','草稿'),('wait_prove','待批'),('proved','已批'),('rejected','被拒')]`

这表示，请假单有'draft', 'wait_prove', 'proved', 'rejected' 四个状态，'草稿', '待批', '已批', '被拒' 是四个状态的显示名称。

名称:

x_state

字段标签:

状态

字段类型

属性

字段类型:

selection

Object Relation:

NULL

Relation Field:

Field Selection:

[('draft','草稿'),('正在删除','正在删除')]

尺寸大小:

64

手动创建:

自定义字段

领域:

[]

要求:

☐

仅读:

☐

可查询:

不可查询

Translate:

☐

正在删除:

置空(NULL)

图 七-3 增加状态字段

实际软件开发中，好的做法是，state 字段是必须且只读的，其值由系统设定。新建的请假单默认设为“草稿”状态，当点击“送批”按钮时，变为“待批”，主管批准后变为“已批”，拒绝的话则变为“被拒”。不过，在 OpenERP 的界面操作中，不能指定默认值，另外，为了便于演示和学习，state 不指定为必须和只读。通过代码开发对象和工作流时，可以指定字段的默认值。

修改视图

按前一章的办法，菜单：管理员 → 自定义 → 用户界面 → 视图，分别编辑“请假单列表”和“请假单表单”，增加显示字段 x_state，下面以表单视图为例说明。

- 1) 视图结构中增加”<field name=“x_state”/>”，
- 2) 表单视图中增加“送批”，“批准”，“拒绝”三个按钮。表单视图的视图结构中增加下述内容：

```
<separator colspan="5" string="审批流程"/>
<group col="5" colspan="5">
  <field name="x_state"/>
  <button string="送批" name="action_toprove" states="draft"/>
  <button string="批准" name="action_prove" states="wait_prove"/>
  <button string="拒绝" name="action_reject" states="wait_prove"/>
</group>
```

这里增加了三个按钮，分别对应工作流的三个迁移的触发事件。另外，为了视图显示漂亮点，增加了 separator, group 几个视图元素。关于视图可用的元素及用法，请参考官方文档。这里特别注意的是，button 中的 states 属性，这个属性表示在哪些状态下显示本按钮。states=“draft,wait_prove”表示，只有状态为“draft”和“wait_prove”时显示本按钮，多个状态间以逗号隔开。

视图

查找 表单 日历 甘特图 图形 流程

新建 编辑 复制 删除

<< 第一页 < 上一页 [2/129] 下一页 > 最后

视图名称: 请假单表单
对象: x_qingjd
优先级(0=紧急): 2

视图类型: 表单
子栏位:
继承视图:

视图结构:

```
<?xml version="1.0"?>
<form string="请假单">
  <field name="x_shenqr"/>
  <field name="x_tians"/>
  <field name="x_kaisrq"/>
  <field name="x_shiyou"/>
  <separator colspan="5" string="审批流程"/>
  <group col="5" colspan="5">
    <field name="x_state"/>
    <button string="送批" name="action_toprove" states="draft"/>
    <button string="批准" name="action_prove" states="wait_prove"/>
    <button string="拒绝" name="action_reject" states="wait_prove"/>
  </group>
</form>
```

图 七-4 视图增加按钮

二、 创建工作流、活动和迁移

工作流和活动

菜单: 管理员 → 自定义 → 工作流定义 → 工作流, 新建工作流。

- **名称:** 请假申请工作流
- **资源对象:** x_qingjd, 即请假单对象。资源对象指定本工作流操作的数据对象。
- **待建 (on_create):** 当系统新建一个工作流对应的资源对象时, 是否同时创建该资源对象的工作流实例。默认是 True, 因而资源对象的每一个实例都有一个对应的工作流实例, 工作流实例保存在数据表 wkf_instance 中。
- **活动:** 新建本工作流的四个活动: draft, wait_prove, proved, rejected, 活动名称不一定要和对象状态名一致, 不过, 为了方便, 这里取一样的名称。

工作流的 XML 定义格式如下。

```
<?xml version="1.0"?>
<terp><data>
  <record model="workflow" id=workflow_id>
    <field name="name">workflow.name</field>
    <field name="osv">resource.model</field>
    <field name="on_create">True | False</field>
  </record>
</data></terp>
```

工作流

查找 表单 日历 甘特图 图形 流程

保存 保存并编辑 取消

<< 第一页 < 上一页 [1/1] 下一页 > 最后页 >>

名称: 请假申请工作流 资源对象: x_qingjd

待建: ☒

活动

活动

<< 第一页 < 上一页 [1 - 4 共 4] 下一页 > 最后页 >>


名称	工作流	类别	Python动作	服务端动作	流程开始	流程停止
draft	请假申请工作流	功能	write({'x_state':'draft'})		是	否
wait_prove	请假申请工作流	功能	write({'x_state':'wait_prove'})		否	否
proved	请假申请工作流	功能	write({'x_state':'proved'})		否	是
rejected	请假申请工作流	功能	write({'x_state':'rejected'})		否	是

图 七-5 workflow定义画面


活动定义参考画面如下图所示：

- 名称：活动名称，任意指定。
- 工作流：指定本活动所属的工作流，这里是“请假申请工作流”。
- 类别：有 Dummy, Function, Subflow, Stop All 四种，现在的翻译不太正确，Function 应译为“函数”，不是功能。Dummy 表示不执行任何 Python 动作。Function 表示在 Python 动作中执行对象的一个方法，这里的对象自然是工作流操作的数据对象（本例是请假单对象）。Subflow 表示执行“子流程”中指定的流程。Stop All 表示终止工作流，这通常是异常处理，即流程跳到了一个意外状态。
- 服务端动作：这将在后续章节叙述。例如，如果希望请假申请按钮按下后发 Mail 通知主管，发 Mail 的动作即在此处定义。
- Python 动作：当类别是 Function 时，这里指定要执行的对象的方法。最常见的是 write 方法写入对象状态。write 是 OpenERP 对象的基类中定义的方法，因此，所有 OpenERP 都能用。你也可以自定义一个方法，然后在这里调用。例如，你希望“批准”请假单时，在该员工的年假数据表中扣除相应天数，则可以自己写个对象方法实现这个要求。
- 子流程和信号：只用于 Subflow 类型的工作流，从略。
- 流程开始：如果是流程的开始节点，则选上。本例的开始节点是 draft
- 流程结束：如果是流程的结束节点，则选上。本例的结束节点有两个，proved 和 rejected。
- 分叉模式：有三个选项，XOR，OR，AND，默认是 XOR。XOR 表示，由本节点始发的出迁移中，沿着第一个满足迁移条件的迁移跳转。OR 表示由本节点始发的出迁移中，只要满足迁移条件即沿该迁移跳转。AND 表示由本节点始发的出迁移中，只有所有迁移皆满足迁移条件才跳转，而且是同时沿所有迁移跳转。XOR 只有一个跳转，OR 有零或多个跳转，AND 有零或全部跳转。
- 合并模式：有两个选项，XOR，AND，默认是 XOR。XOR 表示，以本节点为终点的入迁移中，只要有一个跳至本节点，即执行本节点的 Action。AND 表示，以本节点为终点的入迁移中，只有所有迁移都已经跳至本节点，才执行本节点的 Action。

注意，不要在活动定义画面中定义迁移（出迁移或入迁移），可能是 OpenERP Webclient 的 Bug，在这里定

 **活动**


名称:

工作流:  类别:

服务端动作:

Python动作:

```
write({'x_state': 'draft'})
```

子流程:  信号 (subflow.*):

流程开始: ☒ 流程停止: ☐

分叉模式: 合并模式:

出迁移

图 七-6 workflow活动定义画面

活动（Activity）定义的完整 XML 格式如下。

```
<record model="workflow.activity" id="" activity_id">
  <field name="wkf_id" ref="" workflow_id""/>
  <field name="name">"activity.name"</field>:
  <field name="split_mode">XOR | OR | AND</field>
  <field name="join_mode">XOR | AND</field>
  <field name="kind">dummy | function | subflow | stopall</field>
  <field name="action">"(...)"</field>
  <field name="signal_send">"(...)"</field>
  <field name="flow_start">True | False</field>
  <field name="flow_stop">True | False</field>
</record>
```

迁移

菜单：管理员 → 自定义 → workflow定义 → 迁移，新建迁移：draft → wait_prove, wait_prove → proved, wait_prove → rejected 三个迁移。

迁移

查找 表单 日历 甘特图 图形 流程

基本查询 高级查询

源活动: 目标活动:

过滤 删除 编辑 新建

<< 第一页 < 上一页 [1 - 3 共 3] 下一页 > 最后一页 >>

<input type="checkbox"/>	源活动	目标活动	信号 (按钮名)	要求的角色	条件	触发器对象	触发条件式
<input type="checkbox"/>	draft	wait_prove	action_toprove		True		×
<input type="checkbox"/>	wait_prove	proved	action_prove		True		×
<input type="checkbox"/>	wait_prove	rejected	action_reject		True		×

图 七-7 工作流迁移

迁移定义画面如下:

- 源活动: 迁移的起始活动
- 目标活动: 迁移的终止活动
- 条件: 是否迁移的检查条件, 是一个表达式或对象的一个方法。表达式中可以使用的变量可以是常量或对象的字段名, 表达式例子有: True, 'hello' in ['hello','bye'], zip==1400, phone==mobile。
- 信号: 触发迁移的按钮名, 这个名字和视图中的 button 的 name 属性一致。
- 要求的角色: 只有该角色的操作者才能触发本迁移, 这里暂不考虑角色, 故不填。
- 触发器对象和触发条件式: 触发条件式是一个表达式或本工作流对应的资源对象的一个方法, 触发条件式应返回触发器对象类型的值。通常在非按钮触发迁移时, 检查触发器对象的触发条件式 (方法), 如果返回 False 则不触发迁移。例如, 工作流前面的步骤如果创建了某资源对象, 则沿着本迁移跳转否则不跳转, 则可以用触发条件式检查是否创建了该资源对象。

迁移

查找 表单 日历 甘特图 图形 流程

保存 保存并编辑 取消

<< 第一页 < 上一页 [2/3] 下一页 > 最后一页 >>

源活动: [wait_prove](#) 目标活动: [proved](#)

条件: True

信号 (按钮名): [action_prove](#) 要求的角色:

触发器对象: 触发条件式:

图 七-8 工作流迁移定义画面

迁移的完整 XML 定义格式如下。

```
<record model="workflow.transition" id="transition_id">
  <field name="act_from" ref="activity_id'_1_'" />
  <field name="act_to" ref="activity_id'_2_'" />
  <field name="signal">(...)</field>
  <field name="role_id" ref="role_id'_1_'" />
  <field name="condition">(...)</field>
  <field name="trigger_model">(...)</field>
  <field name="trigger_expr_id">(...)</field>
</record>
```

90

```
</record>
```

至此，“请假申请 workflow” 定义完毕，下面测试该 workflow。

三、 测试 workflow

回到主菜单，点击 请假申请 → 请假单，新建一张请假单，如下图所示：
在这个表单中，和第二章相比，多了个“审批流程”的分界线，这正是 separator 标签的作用。还多了状态和按钮。状态为“草稿”时，显示“送批”按钮，“待批”状态时，显示“批准”和“拒绝”两个按钮，其他状态时，不显示任何按钮。点击按钮时，状态自动变化。状态的变化，正是活动定义中的 Python 动作执行的结果。



图 七-9 请假申请 workflow 测试画面

本例的 workflow 还有一些问题，一是应该禁止修改状态，状态只能由系统改变。二是应该只有主管才能点击“批准”和“拒绝”按钮。这两个问题会在后续章节解决。

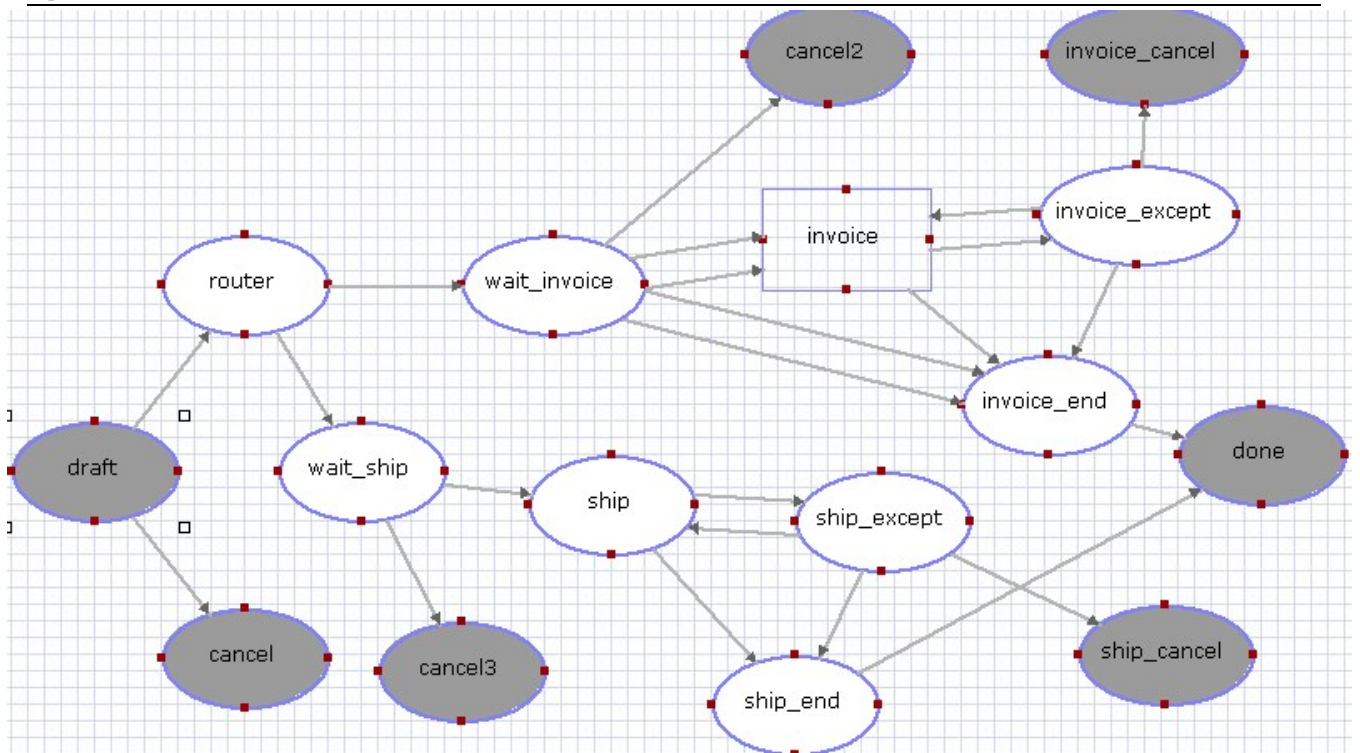
第三节 复杂 workflow 实例分析

本节内容有些复杂，建议初学者跳过，以后再学习。本节以销售订单（Sale Order）的内部处理流程为例，说明复杂 workflow 的实现机制。在 OpenERP 中新建一个销售订单时，订单状态为 draft。确认（Confirm）订单时，流程分叉，一支进入财务票据（Invoice）处理，等待财务确认并开发票；一支进入出库处理，等待仓管员确认并出库。两个分支都结束了的话，销售订单处理结束。任何一支处理出错或取消的话，流程异常结束。

点击销售订单表单画面左下角的 Customise → Manage Workflows，可以打开销售订单处理 workflow (sale.order.basic)，如下图所示。

本节的分析，涉及的源代码文件有：

- server\addons\sale\sale_workflow.xml，workflow sale.order.basic 的定义文件。
- server\addons\sale\sale.py，workflow 节点的 Action 方法的定义文件。
- server\addons\mrp\mrp_workflow.xml，workflow mrp.procurement.basic 的定义文件。
- server\addons\mrp\mrp.py，mrp_procurement 对象及 workflow 节点 Action 定义文件。



- 1) 销售订单处于 draft 时，如果点击“Cancel”按钮，工作流进入“Cancel”节点，订单处理结束；如果点击“Confirm”按钮，进入“router”节点，继续处理。
- 2) 看看 router 节点有两个分叉，其分叉模式定义是：<field name="split_mode">OR</field>，即，如果条件满足，会同时进入两个分支。实际上，两个分支都是无条件跳转（参见迁移 trans_router_wait_invoice 和 trans_router_wait_ship）。到 wait_invoice 的分支暂且不表，且看到 wait_ship 的分支。
- 3) wait_ship 节点又有两个分支，如果此时按下“Cancel”按钮，处理结束。到 ship 的分支定义如下：

```
<record id="trans_wait_ship_ship" model="workflow.transition">
  <field name="act_from" ref="act_wait_ship"/>
  <field name="act_to" ref="act_ship"/>
  <field name="condition">(order_policy!='prepaid') or invoiced</field>
</record>
```

条件 (order_policy!='prepaid') or invoiced 表明，如果本订单要求先付款再发货，则流程停在 wait_ship，等待财务开好票，才会进入 ship 分支。

- 4) 再看 ship 节点，该节点的两个出迁移定义如下：

```
<record id="trans_ship_ship_end" model="workflow.transition">
  <field name="act_from" ref="act_ship"/>
  <field name="act_to" ref="act_ship_end"/>
  <field name="trigger_model">mrp.procurement</field>
  <field name="trigger_expr_id">procurement_lines_get()</field>
  <field name="condition">test_state('finished')</field>
</record>
<record id="trans_ship_ship_except" model="workflow.transition">
  <field name="act_from" ref="act_ship"/>
  <field name="act_to" ref="act_ship_except"/>
  <field name="condition">test_state('canceled')</field>
</record>
```


ship 节点的分叉模式是默认的 XOR，也就是说，不会同时跳入两个分支，只会进入一个分支。注意第一个分支的 trigger_model 和 trigger_expr_id，这表示，系统将调用方法 procurement_lines_get()，该方法在文件 sale.py 中定义，它返回订单明细行上的 procurement 对象 (procurement_id)。根据 act_ship 节点的 action “action_ship_create”，每一个订单明细行都应该对应的创建了一个 mrp_procurement 记录。如果没有，说明 workflow 异常，不应进入 act_ship_end 节点，而是继续检查第二个分支 trans_ship_ship_except，无条件进入 act_ship_except，即发货异常。

第四节 企业流程(Enterprise Process)

在 OpenERP 中企业流程(Process)是一种视图，也就是说，是一种显示界面，不是真正的流程。真正的流程是 Workflow。

认识企业流程

在第一章的案例中，点击菜单：销售管理 → 销售订单 → 新报价单，创建一张报价单。在报价单表单的右上方，有个“流程(Process)”的按钮，点击它，看到如下的销售过程流程图。

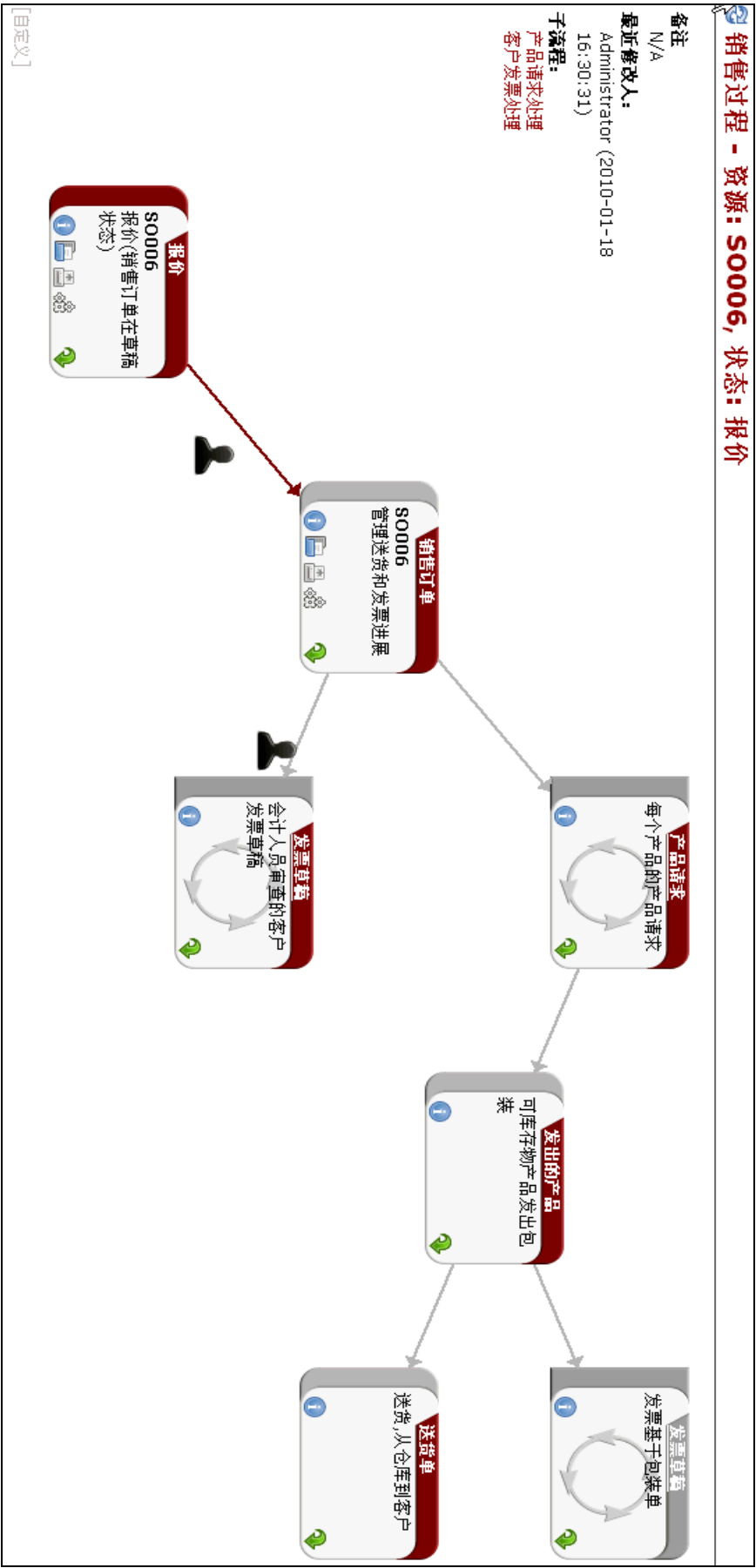


图 七-10 企业流程示意图

在这张图上，汇集了销售关联的主要步骤和菜单功能。通过这张图，有关人员能迅速了解企业的销售

业务流程，也能迅速找到销售业务各个步骤的操作菜单。实际上，这两点正是 OpenERP 的企业流程主要作用。

在企业流程图中，当前节点（活跃节点）的左边框是红色的，如上图的第一个节点，“报价”节点。非活跃节点的左边框是灰色的，如上图中其他各个节点。不可达节点的左和上边框都是灰色的，如上图中的发票草稿。活跃节点的意思是，当前对象，本例是新建的草稿状态的报价单，正处于该节点下。定义企业流程时，每一个流程节点（或说 Action）都有一个关联的对象（Object）和对象的状态。如果当前操作的对象及对象状态和节点关联的对象及状态一致，则该节点处于活跃状态。反之则是非活跃状态。不可达状态是，以当前的活跃节点为起点，不可能到达该节点。

对于活跃节点，节点上有多个按钮可以操作。从左至右，依次是：

- 帮助：打开和该节点关联的 Help URL。
- 打开（Open）：打开该节点关联的对象的表单。
- 打印（Print）：列出和该节点的关联对象相关联的所有报表，供选择打印。对象关联的报表，通常也显示在该对象的表单视图的右边工具条上。此处列出的报表选项和视图右边工具条上显示的是一样的。
- 打印工作流（Print Workflow）：如果节点关联的对象有对应的工作流，则显示本按钮，打印该工作流。
- 关联菜单：右边那个绿色跳转箭头，定义流程节点时指定的操作菜单。

实际上，活跃节点上的各个按钮功能，在关联对象的表单或列表视图上都有同样功能。也就是说，企业流程只不过是流程图的方式，把相关功能集成到一起，方便操作和查看，并不增加任何新功能。所以说，企业流程和表单视图一样，是一种不同形式的显示视图。而工作流（Workflow）是改变对象状态的一项功能，而不是用于显示的视图。不过，企业流程和其他视图不同的是，企业流程中的每个节点都可以和不同的对象关联，因此，企业流程图中可以同时呈现多个对象的操作菜单。不象表单视图，只能显示一个对象及其关联功能（报表、Wizard 等）。

企业流程在菜单：管理员 → 自定义 → 企业流程 中定义。定义方法和工作流类似。

第八章 OpenERP 的报表开发

第一节 OpenERP 报表开发的预备知识

OpenERP 报表有两种，一种是基于 RML 的，一种是基于 XSLT+XML 的。开发 RML 报表的时候，需要手工做好 RML 文件，RML 文件类似于 HTML 文件，它定义报表的显示格式。当我们做好了 RML 文件的时候，希望有个工具，能把 RML 转换为 PDF 文件，预览报表效果。基于 XSLT+XML 的报表，在 OpenERP 的内部处理中，是先把 XSLT+XML 转换为 RML 文件，再转换成 PDF 报表。开发中，我们也需要工具转换 XSLT+XML 成 RML，再转换成 PDF 文件，以预览报表效果。本书附件的 source/ch03 目录中，带有 xsl2rml.py 和 rml2pdf.py 两个工具，用于上述转换。

一、 报表开发工具

报表工具安装

为了使用 xsl2rml.py 和 rml2pdf.py 两个工具，需要安装一些 Python 模块。

1) 安装 Python

安装 OpenERP 的时候，Python 已经装好了，不过，3.0 及以上版本支持软件较少，建议不安装。本书安装的是 2.6.2 版本，安装 2.5.x 版应该也没问题。安装 2.5.x 版的话，相关包也要安装 2.5 版的。

2) 安装 python 的 pdf 开发包

PDF 开发包有如下两个，RML 转换成 PDF 的工具需要这两个包。

PIL-1.1.6.win32-py2.6.exe

reportlab-2.3.win32-py2.6.exe

3) 安装 python 的 xslt 及 xml 处理包

xslt 及 xml 处理包有下述两个，xsl2rml.py 工具需要依赖这两个包。

libxml2-python-2.7.4.win32-py2.6.exe

lxml-2.2.2.win32-py2.6.exe

上述软件全是开源软件，可以在相关网站下载。

xsl2rml.py 和 rml2pdf.py 工具用法

xsl2rml.py 和 rml2pdf.py 工具参见本书附件的 source/ch03 目录。工具功能尚很弱，如果 xsl、rml 文件有语法错误，工具检查不出来，只是异常退出。所以发生错误时，请首先检查文件语法错误。下面列举该工具的用法。

1) Xsl + xml => rml, 这个例子就是下文 XSL 简介的例子，如下：

```
cd source\ch03\report
xsl2rml.py example\ex1.xsl example\ex1.xml example\ex1.rml
```

xsl2rml.py 是一个简单的工具，它合并 xsl 文件和 xml 文件，转换成 RML 文件。用法是：

```
xsl2rml.py .xsl 文件 .xml 文件 .rml 文件
```

参数中的.RML 文件指定要输出的 RML 文件名，前两个参数是输入参数。

2) rml => pdf

```
cd source\ch03\report
rml2pdf\rml2pdf.py example\ex2.rml
```

这个工具将参数中的 RML 文件转换成 PDF 文件，输出的 PDF 文件和输入的 RML 文件同路径同文件名，后缀是 PDF。

3) 更多例子参考 source\ch03\report\example 下的例子。

二、 XSL 简介

XSL 是样式描述语言，通过 xslt 引擎，将 xml 数据和 xsl 样式合并，能生成 html, xml, pdf 等格式文档。关于 xsl 的知识，参考 XSLTutorial, XSLTutorial 非常浅显易懂。XSLTutorial 的下载地址：<http://www.w3.org> 下面的例子来自 XSLTutorial:

在 XSL 中定义输出格式，在 XML 中定义数据，XSL 使用 XPath 引用 XML 中的数据，如下图中的 <xsl:value-of select="//title"/>, "//title" 引用 XML 中的<title>标签中的值。通过 XSLT 引擎，合并 XSL 和 XML 文件，输出指定格式的文件。可以在 XSL 引擎中指定输出格式，常见的格式有 HTML、TXT、XML、PDF 等，不同的引擎支持的输出格式种类有所不同。

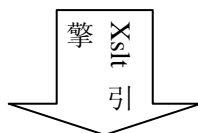
ex1.xsl:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version = '1.0'

xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match="/">
<h1><xsl:value-of select="//title"/></h1>
<h2><xsl:value-of select="//author"/></h2>
</xsl:template>
</xsl:stylesheet>
```

ex1.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<source>
<title>XSL</title>
<author>John Smith 先生</author>
</source>
```



```
<h1>XSL</h1>
<h2>John Smith 先生</h2>
```

图 八-1 XSLT 样式处理示意图

第二节 OpenERP 报表简介

Open ERP 的报表有两种，一种是基于 Openoffice 模版的，一种是基于 xslt + xml 的。前者占多数，格式复杂些的报表基本上都是前者，后者很少，基本上都是简单报表。

一、 基于 sxw 或 odt 模板的报表

基于 Openoffice 的报表处理过程是，通过 OpenOffice 编辑报表格式，这和 Word 编辑表格差不多。在需要插入动态数据的地方，插入[[python 表达式]]，OpenERP 生成报表时，会计算[[]]里的表达式的值，填入相应位置。编辑好的文档存成 OpenOffice 的 sxw 格式，再通过 SXW2RML 工具转换成 RML 格式的文件。从 OpenOffice 文档到 RML 文件，不属于 OpenERP 的处理，必须手工处理或借助其他工具处理。

从 RML 开始，OpenERP 读入 RML 文件，计算其中的[[]]里的表达式，利用 ReportLab 库，把 RML 文件转换成 PDF 报表文件。

但是，目前没有免费的 SXW2RML 工具可用，仅有收费的 ReportDesigner 工具支持 SXW2RML。因此，实际上，用 OpenOffice 做的报表模版，如果不利用收费软件，没法供 OpenERP 使用。实践上可行的办法只好是，手工编辑 RML 文件。RML 类似于 HTML，用于定义 PDF 文档的显示格式，在里面嵌入[[]]形式的 Python 表达式，即可访问 OpenERP 对象，从数据库取得动态数据，生成报表。

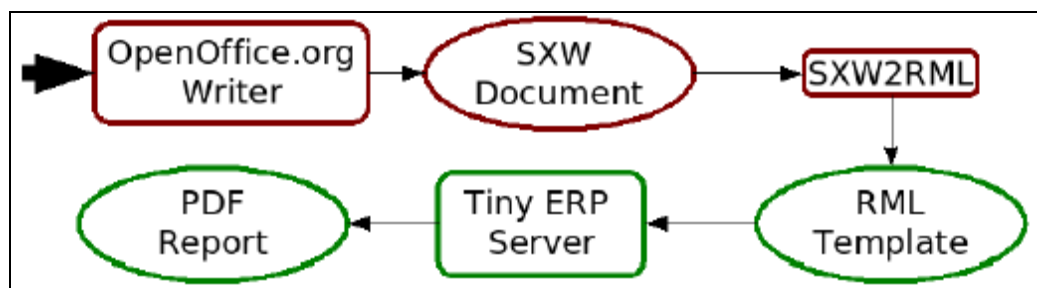


图 八-2 SXW 报表处理过程

本类型报表的开发步骤：

1) 编辑 RML 报表模版文件

RML 定义文件的 Tag 及语法和 HTML 相似，详细说明参见文档 rml-reference.pdf。具体例子参考：base/module/report/ir_module_reference.rml。点击菜单 管理员 → 模块管理 → 模块，选择一个模块，点击右边的工具条上的报告下的“技术指南”，即可看到该报表的显示效果。



RML 语法参考文档下载地址：<http://www.shine-it.net/viewthread.php?tid=713&page=1&authorid=593>

openobject-developer.pdf 下载地址：<http://doc.openerp.com/>

2) 预览模版文件格式

使用本书附件的 source/ch03 目录下的工具 rml2pdf.py 转换 rml 文件成 pdf 文件，预览模版文件格式是

否正确。rml2pdf.py 的用法后文另述。

3) RML 模版中插入 Python 代码（用[[]]括起）

模版中插入 python 代码取得动态数据，具体说明和例子参考文档 openobject-developer.pdf，以及例子 base\module\report\ir_module_reference.rml。

4) 编写报表对象代码

编写报表对象定义的 py 文件，详细说明和例子参考文档 openobject-developer.pdf，以及例子 base\module\report\ir_module_reference_print.py。编写好代码后注意不要忘记在模块的 __init__.py 中添加导入报表对象的 import 语句。

5) 编辑报表定义的 xml 文件

详细说明和例子参考文档 openobject-developer.pdf，以及例子 base\module\module_report.xml。也可以通过界面操作导入，操作菜单：管理员 → 低级对象 → 操作 → 报表 XML，在该界面上指定报表 RML 文件，报表定义对象，报表存取动态数据的对象等。

6) 发布

升级模块，做成的报表即可生效。当浏览或编辑报表对应的数据对象时，画面右边即会出现打印对应报表的菜单。

二、 基于 xsl+xml 模板的报表

基于 XSL+XML 的报表处理过程是，OpenERP 读入 XML 文件，根据 XML 文件的指示，调用相应对象从数据库取得数据；再读入 XSL 文件，XSL 文件定义了报表的格式。将数据合并到报表格式中，生成 RML 文件，最后调用 ReportLab 把 RML 文件转换成 PDF 文件。和前一种报表不同的是，不是通过嵌入的 Python 语句取得动态数据，而是将取得数据的命令编入 XML 文件。报表格式是通过更具通用性的 XSL Style sheet 来定义。

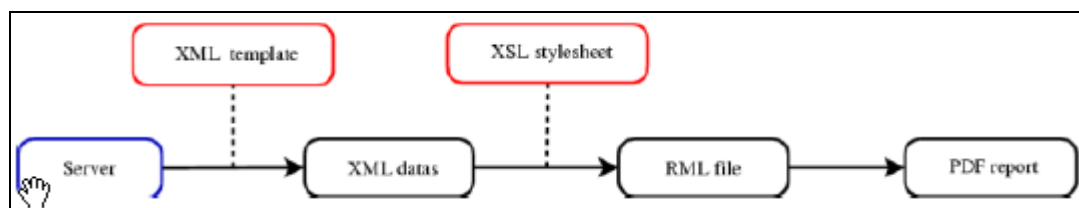


图 八-3 xsl+xml 报表处理过程

本类型报表的开发步骤是：

- 1) 编辑报表的 xslt 样式文件，定义报表的格式。
- 2) 编辑取数据的 xml 文件。
- 3) 选择菜单“管理员->低级对象->操作->报表 xml”，在报表 xml 里新建报表，指定 xsl 和数据 xml 的存放路径，则报表定义完成。
- 4) OpenERP Server 运行时，根据数据 xml 文件取得数据，合并数据和 xsl 样式文件，生成 rml 文件，再从 rml 文件生成 pdf 报表文件。

详细开发方法参考“openobject-developer.pdf”的第 14 章。OpenERP 的实际例子参考 base 模块中的“打印名片”报告，对应报表文件是 base\res\partner\report\partner_address.xsl、base\res\partner\report\

partner_address.xml、base/res/partner/partner_report.xml。

第三节 报表中文化解决方法

目前的 OpenERP 报表中，主要存在这么几个问题：

- 1) 不能显示汉字（汉字在 PDF 报表中显示成黑色方块）
- 2) 报表中的英文需要翻译成中文
- 3) 报表中的地址、日期等显示格式不符合中文习惯

下面依次给出上述几个问题的解决办法。

一、让报表显示汉字

先说第一个问题，汉字之所以会显示黑色方块，原因是 OpenERP 生成报表时，ReportLab 找不到中文字对应的字体。ReportLab 中支持多种非亚洲字体库，但亚洲字体比较复杂，而且字数多，目前 ReportLab 支持的亚洲字体很少。OpenERP 自带的 Reportlab 库目前支持一种中文字体“STSong-Light”，但 OpenERP 调用 ReportLab 的时候，没有注册（初始化）该字体。

明白了原因，解决办法就出来了。一是在 OpenERP 中调用 ReportLab 时，注册字体“STSong-Light”，二是，将报表的 RML 文件和 XSL 文件中的字体全部替换成“STSong-Light”。具体方法如下：

修改报表模块代码。

在“server\bin\report\render\rml2pdf__init__.py”中增加下述三行代码，注册中文字体“STSong-Light”。

```
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.cidfonts import UnicodeCIDFont
pdfmetrics.registerFont(UnicodeCIDFont('STSong-Light'))
```

对于 OpenERP windows 版，修改上述代码，用 python2.5 编译 server\bin\report，编译命令参考如下：
注意，必须用 Python2.5 版编译，否则 OpenERP 不能启动。

```
PATH=C:\Python25;%PATH%
cd server\bin\
python -O -m compileall .\report
```

编译好以后，替换 OpenERP server 安装目录下的“Server\library.zip”中的 report 模块，重新压成 library.zip 文件，替换原来的 library.zip 文件，重启 OpenERP Server 即可。或者，从这里下载编译好的 Python2.5 版的__init.py__文件：

<http://www.shine-it.net/viewthread.php?tid=713&page=1&authorid=593>

对于 Linux 版，将修改好的__init.py__代码覆盖“site-packages/openerp-server/report”中同名文件，重启 OpenERP Server 即可。

将所有模块的 xsl 和 rml 文件中的字体替换成“STSong-Light”。

在 Windows 上，可以通过文本编辑工具自动搜索所有 xsl 和 rml 文件，并替换其中的字体。要替换的字体有下面一些：

Times-Roman

Times-BoldItalic
Times-Bold
Times-Italic
Helvetica-BoldOblique
Helvetica-Bold
Helvetica
Courier-Bold
Courier

在 linux 上，可用本书附件的下述脚本一次性批量修改。

source\ch03\report-cn.sh

不过，报表头和尾的样式不定义在任何 RML 和 XSL 文件中，而是直接嵌在代码中。因此，还必须修改代码中的报表头和尾的样式中的字体。否则报表头和尾中的公司名等汉字仍会显示成方块。OpenERP 中的报表头和尾（页眉/页脚）定义在“addons\base\res\res_company.py”的“_get_header”和“_get_header2”中。将其中的字体改成“STSong-Light”，重启 OpenERP Server，升级 base 模块，则修改生效。或者，如果不想升级 base 模块，也可以直接在数据库中改字体。代码“_get_header”和“_get_header2”中的内容在安装 base 模块时加载到了数据表 res_company 的字段 rml_header2 中，直接修改其中的字体即可。

二、 让报表说中文

对于报表中的标题等英文，直接在 XSL 或 RML 文件中翻译成中文即可。对于地址等的显示习惯，需要修改 RML 或 XSL 文件中的显示格式。例如“base\res\partner\report\partner_address.xml”中的名片打印，地址按外国习惯是“门牌号，街道，城市，省，国”，必须改成中国习惯的“国，省，市，街道，门牌号”。这只要调整 xsl 中的地址的各字段的显示顺序即可。

第四节 实作报表

本节实作打印请假单的 PDF 报表，请假单报表基于 RML 格式。根据前面的介绍，RML 格式报表的大致开发步骤是，手工作成 RML 文件，编写报表对象代码，编写导入报表的 XML 文件。

一、 编写 RML 文件

要实作的请假单格式如下：

[[repeatIn(objects, 'QingJD')]]

OpenERP应用和开发基础 - 请假单

请假人	[[QingJD.x_shenqr.name]]		
请假天数	[[QingJD.x_tians]]	开始日期	[[QingJD.x_kaisrq]]
请假事由	[[QingJD.x_shiyou]]		
审批状况	[[QingJD.x_state]]		

申请日期: [[QingJD.create_date]]

图 八-4 请假单格式

做好的 RML 文件参见本书附件的 source\ch03\ch03report\rpt_qingjd.rml。要预览报表效果，命令如下：

```
cd source\ch03\
report\rml2pdf\rml2pdf.py ch03report\rpt_qingjd.rml
```

二、 编写报表对象代码及 XML 文件

报表定义对象代码如下，文件参见 source\ch03\ch03report\rpt_qingjd.py

```
# -*- encoding: utf-8 -*-
import time
from report import report_sxw
from osv import osv

class rpt_qingjd(report_sxw.rml_parse):
    def __init__(self, cr, uid, name, context):
        super(rpt_qingjd, self).__init__(cr, uid, name, context)
        self.localcontext.update({
            'time': time,
        })
report_sxw.report_sxw('report.ch03report.rpt_qingjd', 'x_qingjd', 'addons/ch03report/rpt_qingjd.rml', parser=rpt_qingjd, header=False)
```

报表定义对象代码基本上所有报表都是一样的，只有代码中红色部分不一样。其中，

- class **rpt_qingjd**: 这是 Class 名，可以任意取名
- **report.ch03report.rpt_qingjd**: 这是本对象的内部名字，可以任意取，以“report.”开头。
- **x_qingjd**: 和报表关联的数据对象，报表中的数据通过该对象访问，本例是请假单对象。
- **addons/ch03report/rpt_qingjd.rml**: 和报表关联的 RML 文件名及路径。
- parser=**rpt_qingjd**: parser 就是本报表 Class。
- header=False: 表示要不要在本报表中显示公司统一的表头和表尾。

导入报表的 XML 文件如下，文件参见 source\ch03\ch03report\rpt_qingjd.xml

```
<?xml version="1.0" encoding="utf-8"?>
<openerp>
  <data>
    <report id="report_ch03report_rpt_qingjd" header="False"
      auto="False" model="x_qingjd" name="ch03report.rpt_qingjd"
      rml="ch03report/rpt_qingjd.rml" string="打印请假单"/>
  </data>
</openerp>
```

其中，

- **model**：和本报表关联的数据对象，这里是请假单对象
- **name**：本报表的报表定义对象名，是报表 PY 文件中的”**report.ch03report.rpt_qingjd**”的”report.”以后的部分。
- **rml**：和本报表关联的 RML 文件
- **string**：界面上右边工具栏上显示的本报表的对应菜单。

三、 测试

将本书附件中 source\ch03\ 下的 ch03report 目录拷贝到 OpenERP 的 addons 下，安装模块 ch03report 。回到主菜单，进入请假单表单，右边工具条上可看见“打印请假单”的菜单。如下图：

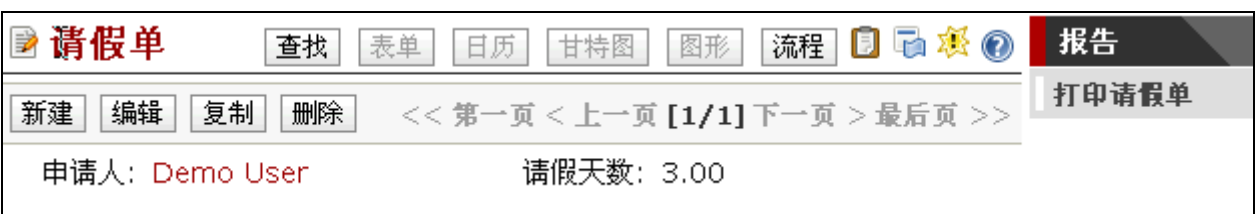


图 八-5 打印请假单

点击请假单，可看到如下 PDF 报表：

75%

搜索网络

OpenERP应用和开发基础 - 请假单

请假人	Demo User		
请假天数	3.00	开始日期	2009-12-30
请假事由	为了提高工作效率，提高工作水平，为老板您创造更多价值，我特此请假学习《OpenERP应用和开发基础》，望您批准。		
审批状况	待批		

申请日期：2009/12/30

图 八-6 请假单报表

第九章 Request,Sequence and Scheduler

本章介绍 OpenERP 开发中的几个有用的小机制：Request,Sequence and Scheduler。Request 用于用户间互发消息，或者系统向用户发送消息。Sequence 是序号生成器，开发一些单据时，经常需要根据一定规则生成单据号，Sequence 可以配置序号生成规则。Scheduler 可以设置一些自动执行的 Job。

第一节 Request 和 References

一、Request

什么是 Request？简单的说，OpenERP 的 Request 就是在线消息。举例来说，你通过 OpenERP 提交了一张请假单，你希望主管快点审批。这时，你可以使用 OE 给主管发送一个 Request，Request 里写明你的请求。如果请假模块是你自己开发的，当有人提交请假单时，你也可以在模块中自动发送 Request 给主管，并且以附件的形式添加请假单到 Request 中。当主管登录 OE 时，他能立即看见 Request，主管可以回复 Request，或者关闭 Request。OE 会自动记录 Request 的发送、回复、关闭等操作历史。

Request 的操作。在 Web Client 的右上角，或者 GUI Client 的“用户”菜单下，可以发送、接收、查看 Request。



点击 Request，进入 Request 的列表画面，继续点击，进入 Request 的表单画面。

Requests

Search Form Calendar Gantt Graph Process

Basic Search

Advanced Search

From: To: Subject:

Filter Delete Edit New

<< First < Previous [1 - 1 of 1] Next > Last >>

	Created Date	From	To	Subject	Priority	Trigger Date	State
<input type="checkbox"/>	04/11/2010 18:12:28	系统管理员	系统管理员	Procurement calculation report.	Normal		draft x

- **From:** 发送人;
- **To:** 接收人, 接收人登录 OE 时, 会看到消息提示 (注意, 消息提示不是即时显示, 而是在登录系统时显示);
- **Subject:** 消息主题;
- **Request:** 消息内容;
- **Trigger Date:** 系统当前未使用, 仅供显示, 没其它特别含义;
- **References:** 相当于附件, 可以引用其它对象的数据 (资源);
- **Status:** 有 ('draft','draft'),('waiting','waiting'),('active','active'),('closed','closed') 几个状态。

Request 对象的源代码参考 server\bin\addons\base\res\res_request.py。

二、 References 字段类型

Request 的代码中 (res_request.py), 两个 Reference 字段的定义如下。

```
'ref_doc1':fields.reference('Document Ref 1', selection=_links_get, size=128, states={'closed':[( 'readonly', True )]}),
'ref_doc2':fields.reference('Document Ref 2', selection=_links_get, size=128, states={'closed':[( 'readonly', True )]}),
```

References 字段, 表示引用别的对象的资源 (或者说, 引用别的数据表的一条记录), 数据库中的存储形式是 (对象名, ID), 如 (product.product,3), 表示引用对象 product.product (数据表 product_product) 中 id=3 的数据。References 字段在界面上的显示通常是, 一个下拉框, 用于选择对象名, 一个查找框, 用于选择该对象的具体记录。记录在界面上显示的是 name, 不是 ID。如(product.product,3), id=3, 但显示的是 “[PC1] Basic PC”, 这是 id=3 的 product 的 name。

上述定义中的 “selection=_links_get” 表示, 通过函数_links_get 取得一序列列表, 该序列列表是诸如

这样的东西：[(product.product,Product),(crm.case,Case),(account.invoice,Invoice)]，即（对象名，对象显示名）的列表，该列表的内容即是界面上的下拉框中显示的内容。该列表的内容来自对象"res.request.link"（数据表 res_request_link），对象 res.request.link 的数据来自各个模块，如 (product.product,Product)，在 product 模块的文件 product_data.xml 中，有如下数据导入代码。

```
<record id="req_link_product" model="res.request.link">
    <field name="name">Product</field>
    <field name="object">product.product</field>
</record>
```

该代码导入 (name=Product, object=product.product) 的记录到数据表 res_request_link 中，这样，Request 中便可以引用 Product 对象的数据。如果希望 Request 中可以引用请假单对象，则请假模块中必须将请假单对象导入到 res_request_link 中。

代码中 states={'closed':['readonly',True]} 的意思是，Request 的状态为 closed 时，本字段只读。

第二节 Sequence

OpenERP 的 Sequence，是序号生成器，类似于数据库中常见的 id 生成器。例如，发票和会计凭证，它们都需要序号，而且，这个序号通常不是简单的数字，而是按一定规则组合的字母和数字。在软件开发中，需要序号的情况很多，OpenERP 的 Sequence 就是满足这种需要的东西。

Sequence 的配置。OpenERP 中，可以添加任意多的序号生成器，每一序号生成器可以配置前缀、后缀、序号间隔等。即序号的一般格式是（前缀序号 后缀），其中，序号为按一定间隔递增的数字。Sequence 配置画面是，Administrator --> Configuration --> Sequences --> Sequences，如下。

Legend (for prefix, suffix)		
Year with century: %(year)s	Day of the year: %(doy)s	Hour 00->24: %(h24)s
Year without century: %(y)s	Week of the year: %(woy)s	Hour 00->12: %(h12)s
Month: %(month)s	Day of the week (0:Monday):	Minute: %(min)s
Day: %(day)s	%(weekday)s	Seconde: %(sec)s

本例中，Account Journal 的序列生成器生成的序列格式形如：1004-001，即前缀为年月 (%(y)s%(month)s-)，序列长度为 8 位（Number padding=8），序列间隔为 1（Increment Number）。

Fiscal Years 是 account 模块添加的。安装 account 模块后，OpenERP 的序列生成器可以按会计年度定义分别采用不同的序列生成器，即不同会计年度，按不同规则采集序列。序列生成器的代码参见：server\bin\addons\base\ir\ir_sequence.py，account 模块对 Sequence 的修改参见：server\bin\addons\account\sequence.py，对 Sequence 配置画面的修改参见：

server\bin\addons\account\sequence_view.xml 。

在 sequence.py 中，继承对象 ir.sequence，在原对象中增加字段 fiscal_ids。该字段为不同的会计年度定义不同的序列生成器。而后，覆盖原对象的序列采集函数 get_id，根据参数给定的会计年度，调用该年度的序列生成器采集序列。如果参数未指定会计年度，或者指定的会计年度没有定义生成器，则仍调用原来的序列生成器。

在 sequence_view.xml 中，继承原来的配置 Form，添加了 Fiscal Years 的 Page，用于配置各会计年度的生成器。

按中国的会计制度，凭证号必须按月编号。一个可取的实现办法是，修改 sequence.py 和 sequence_view.xml，将按年定义生成器改成按月定义生成器，这样就能让凭证的序号生成器每个月都启用新的编号器。

生成新的序列号的代码示例如下：

```
c = {'fiscalyear_id': move.period_id.fiscalyear_id.id}
new_name = self.pool.get('ir.sequence').get_id(cr, uid, journal.sequence_id.id, context=c)
```

本例中，取得凭证的会计年度，从对象池中取得对象 ir.sequence，指定序号生成器 id (journal.sequence_id.id，本例即为 Account Journal 的序列生成器)，调用 get_id 方法取得一个新序号。

第三节 Scheduler

OpenERP 的 Scheduler。软件系统中，经常有些需要定期自动运行的任务。在 unix 中，支持这种需求的是 cron 命令，在 Windows 中，是附件中的任务计划程序。在 OpenERP 中，例如，如果设置了最小库存规则，系统需要每天自动检查产品库存，如果低于设定的最小库存值，则自动产生该产品的供应单。又如，系统每天自动检查收付款条件，到了收付款日，或收付款前后多少天，系统自动产生 Request 或 EMail 提醒相关人。诸如这样的任务，需要一个任务调度器，即 Scheduler。OpenERP 的 Scheduler 就是满足此类需求的东西。

OpenERP 的 Scheduler 的配置画面，Administrator --> Configuration --> Scheduler --> Scheduled Actions，画面如下。

Scheduled Actions

Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [1/2] Next > Last >>

Information Technical Data

Name: Run mrp scheduler Active: ☐

User: 系统管理员 ? Priority: 5

Interval Number: 1 Interval Unit: Days

Next Call Date: 04/11/2010 18:12:09 ? Number of Calls: -1

Repeat Missed: ☐

Scheduled Actions

Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel << First < Previous [1/2] Next > Last >>

Information Technical Data

Action to Trigger

Object: Function:

Arguments:

该配置画面上，可以配置任务每隔多少时间调用一次（Interval Number 和 Interval Unit），下次调用时间（Next Call Date，每次调用任务后，当前时间加上间隔时间，算出下次调用时间），调用多少次后停止调用（Number of Calls），是否重复过去的调用（Repeat Missed，例如，有一段时间服务器未启动，当启动后，是否重复补充未启动期间的作业），自动调用的函数（Technical Data）。本例中，系统每天调用对象 `mrp.procurement` 的方法 `run_scheduler`，参数是 `(False,)`。注意，OE 中方法的前三个参数总是 `(self, cr, uid)`，因此，本例的实际调用形式是 `run_scheduler(cr, uid, False)`。方法 `run_scheduler` 在 `server\bin\addons\mrp\mrp.py` 中。

OpenERP 的 Scheduler 的代码参见：`server\bin\addons\base\ir\ir_cron.py`。其中，函数 `_poolJobs` 实现任务的定期调用，关键代码行是：`self.setAlarm(self._poolJobs, int(time.time()) + next_call, db_name, db_name)`。`ir_cron` 继承自 `netshvc.Agent`，`setAlarm` 定义在 `netshvc.Agent` 中。`netshvc.Agent` 的代码参见 `server\bin\netshvc.py`，其中定义了一个 timer：

```
timer = threading.Timer(wait, fn, args, kwargs)
timer.start()
```

上述代码相当于启动了一个线程，该线程会等待 `wait` 时间后调用 `fn` 函数，这里的 `fn` 是 `ir_cron` 的 `_poolJobs` 方法。关于 `threading.Timer` 的详细解释，参考 Python 的 `threading` 模块说明。

综上分析，`ir_cron` 的实现原理是，每当新建、修改一个定期 Job 的时候，系统第一次调用 `_poolJobs`，`_poolJobs` 启动一个新线程。该线程等待 Job 中设定的间隔时间后激活，激活后再次调用 `_poolJobs`，`_poolJobs` 再次设定下次激活时间，如此反复，即实现了作业的定期调用。

第十章 OpenERP 的权限设置

第一节 OpenERP 权限机制简介

目前，OpenERP 有下述权限设置功能：

1) 菜单、报表访问权限设置

OpenERP 可以设置哪些人可以看见哪些菜单。以及，哪些人能看见和对象关联的右边工具条上显示的报表。例如，请假申请功能中，普通员工没必要看见“请假单审批”菜单，而主管需要该菜单。

2) 对象访问权限设置

设置哪些人可以访问哪些对象，对象的访问权限包括创建、读、写、删除。在 OpenERP 中，对象可以等同于数据表，因此，等同于说，可以设置哪些人拥有数据表的哪些权限（增、删、改、读）。

3) 记录访问权限设置

可以设置哪些人可以访问哪些记录。对象访问权限中设置了对数据表的访问，但并没有设置对记录的访问控制。例如，假如有这么个需求，每个员工只能看见和修改自己的员工信息，主管可以看见和修改自己及下属的员工信息。那么首先，每个员工都要有“员工信息表”的读、写权限，但通常不应该有增、删权限，增、删权限也许控制在人事部手里。其次，每个员工只能访问部分记录，所能访问的记录的查询条件是“自己及自己的下属”的记录。

在 OpenERP 中，可以设置表的查询条件，指定某个组只能访问符合该查询条件的记录。对这些记录的访问权限（增删改读）由对象访问权限设置。

4) 字段访问权限设置

OpenERP 中不仅能设置表和记录的访问权限，还能设置表中的字段的访问权限。例如，员工信息表中，员工能看、但不能修改自己的工资字段，工资字段只能由人事部门修改。这就要求设置，员工组只能读工资字段，人事组可以修改工资字段。

5) 设置 workflow 操作权限

例如，请假申请 workflow 中，“请假审批按钮”只有主管才能点击，普通员工不能点击。OpenERP 中，可以在 workflow 的每一步迁移中，设置哪些角色允许触发本迁移。

上述权限机制，1—4 都是基于组(group)来设置，5 是基于角色(role)来设置。OpenERP 的每个登录用户都属于某些组，以及拥有某些角色。下图是用户及组的设置画面：

操作菜单：管理员 → 用户 → 用户

该画面中，其他各字段的含义都易于理解，只有“默认功能”和“菜单功能”稍稍费解。其中默认功能是该用户登录系统时，执行哪个动作（Action）、显示哪个画面，菜单功能是，在 GUI Client 中，点击窗口上部工具条的“菜单”按钮时，执行哪个动作。画面中“组”和“角色”的 Tab 页分别增加用户所属的组和角色。

用户

查找 表单 日历 甘特图 图形 流程

保存 保存并编辑 取消

<< 第一页 < 上一页 [2/2] 下一页 > 最后一页 >>

姓名: Demo User 有效: ☒

登录名: demo 密码: ●●●●

用户 组 角色

地址: Tiny sprl, Chaussee de Nar 公司: Tiny sprl

默认功能: Menu 菜单功能: Menu

语言: English 时区:

签名: Mr Demo

图 十-1 用户及组设置

第二节 组的权限设置方法

组权限设置

操作菜单：管理员 → 用户 → 组，选择一个组进入。

在这个画面的“菜单”Tab 页中，可以设置组能访问的菜单。在 OpenERP 中，如果没有设置菜单的访问组，则默认所有组都允许访问该菜单。一旦设置了访问组，则只有设置的访问组允许访问该菜单。另外，也可通过操作菜单：管理员 → 安全 → 允许访问菜单，设置哪些组能访问菜单。

在“访问权限”Tab 页，可以设置组能访问的对象及访问权限（增删改读）。也可通过操作菜单：管理员 → 安全 → 访问控制下的“访问控制列表”和“访问控制网格”设置对象访问权限。

组

搜索 表单 日历 甘特图 图表 流程

新建 编辑 复制 删除

<< 第一页 < 上一页 [11/15] 下一页 > 最后一页 >>

组名: Stock / MultiWorker

用户 菜单 访问权限 规则 备注

访问规则

<< 第一页 < 上一页 [1 - 5 共 8] 下一页 > 最后一页 >>

对象	读权限	写入权限	创建权限	删除权限	名称
库存.领料/提货.调拨.向导	是	是	是	是	stock.picking.move.wizard.multi
库存跟踪批次	是	是	是	否	stock.tracking.multi
包装单	是	是	是	是	stock.picking.multi
生产批次	是	是	是	否	stock.production.lot.multi
生产批次次修订	是	是	是	是	stock.production.lot.revision.multi

导入 | 导出 << 第一页 < 上一页 [1 - 5 共 8] 下一页 > 最后一页 >>

图 十-2 对象访问权限设置

在“规则”Tab 页，可以设置组能访问符合哪些条件的记录。这个设置有点复杂。如下图中的组“Stock / MultiWorker”，有多条规则，每条规则定义一个访问的过滤条件。如最后一条规则“限制分公司仅能访问自己公司的 Warehouse”，这条规则对应的需求是“Stock / MultiWorker”组的用户只能访问自己公司或子公司的仓库，规则实现方法是，Warehouse 对象的公司 必须是 当前用户所属公司的子公司。



图 十-3 组访问规则设置

本例规则的设置如下：

- **对象：**规则过滤的对象，即本规则使得，只能访问该对象的符合本规则条件的记录。
- **名称：**任意取
- **隶属：**定义过滤条件，详细格式参照后文的 Domain 表达式。本例设置的含义是，仓库对象的公司（字段“company_id”）是当前用户所属公司（User / Company）的子公司（child_of）。
- **强制范围：**自定义 Domain 表达式，用于过滤条件。含义是只能访问符合本过滤条件的记录。

一条规则可以有多个隶属表达式，多个隶属间是 AND 的关系。一个对象可以有多个规则，这些规则间是 OR 的关系。规则也可通过操作菜单：管理员 → 安全 → 记录规则 设置。

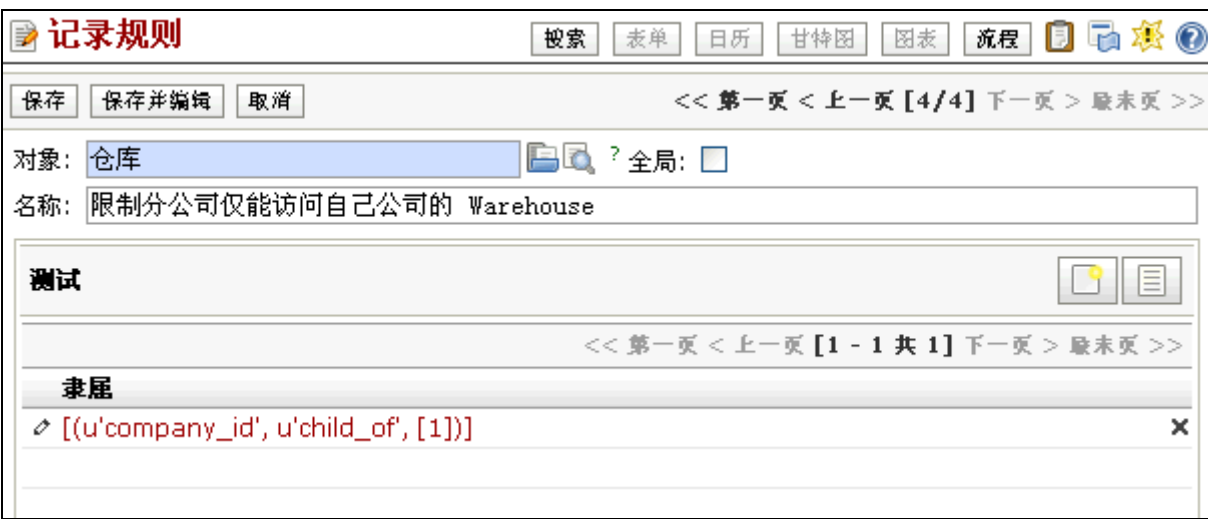


图 十-4 访问规则设置

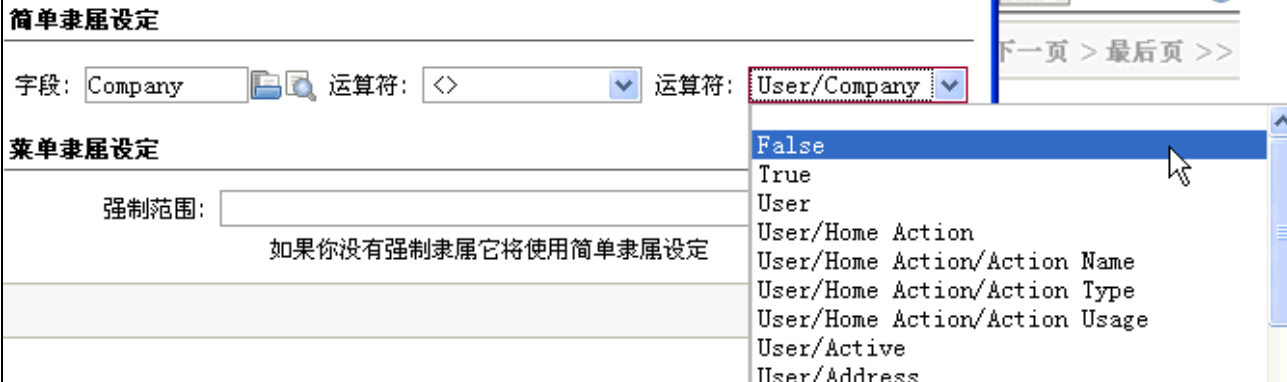


图 十-5 组访问规则过滤条件设置

工作流角色设置

菜单：管理员 → 自定义 → 工作流自定义 → 迁移 中，可以指定触发迁移“要求的角色”。例如，在上一章的“请假申请工作流”中，如果指定了“批准”和“拒绝”要求的角色，则只有拥有该角色的用户点击“批准”和“拒绝”按钮，请假单状态才会变。OpenERP 这里有点 Bug，如果不拥有该角色的用户进来，仍可点击“批准”和“拒绝”按钮，只是点击无效。据说下一个版本（本书实验版本是 5.06）会修改该 Bug，不具有角色的用户将看不见触发按钮。

报表访问权限设置

菜单：管理员 → 低级对象 → 操作 → 报表 XML 中，可以指定允许访问该报表的组。

字段访问权限设置

字段权限设置有两个途径，一是视图中设置，二是对象上设置。视图中设置的方法是，菜单：管理员 → 自定义 → 数据库结构 → 字段，在字段定义画面上有个“组安全性”的 Tab 页，此处指定哪些组可以在视图中看见该字段。也可在视图定义的 XML 中设置，如下：

```
<page string="Extra Info" groups="base.group_admin">
```

这表示，只有 Admin 组才能看见视图上的 TAB 页“Extra Info”。

对象上设置字段访问权限的方法，目前界面操作尚支持不够，只能在对象定义的代码中实现。如下，在对象的字段定义中，增加属性 write 和 read，指定哪些组允许读或写本字段。

```
class res_partner(osv.osv):
    _description='Partner'
    _name = "res.partner"
    _order = "name"
    _columns = {
        'name': fields.char('Name', size=128, required=True, select=True,
            write=['base.group_admin'],
            read=['base.group_admin']),
    }
res_partner()
```

第三节 OpenERP 的预设权限组(Group)

一、 预设权限组定义

OpenERP 的每个模块都预设了一些组，以及那些组对菜单和对象的访问权限。这些预设组对菜单和对象的访问权限，通过“管理员”下的下述菜单查看。

Security

Grant Access To Menus

Access Controls

Access Controls List

Access Controls Grid

“Grant Access To Menus”查看每个菜单项允许哪些组访问，没有指定访问组的菜单，表示任何组都可以访问该菜单。“Access Controls List”显示组对对象的读写删除的访问权限的权限矩阵。“Access Controls Grid”显示每个对象允许哪些组访问。

OpenERP 中的预设组，在每个模块下的 security 目录下的文件：xxx_security.xml、ir.model.access.csv 定义。其中_security.xml 文件定义组和组对菜单的访问权限，ir.model.access.csv 定义组对对象的权限矩阵。如 product 模块，product/security 目录下的文件：product_security.xml 的下述语句，定义了组“Product / Manager”。这个语句的意思是，导入一条记录到数据表 res_groups(model="res.groups", 表示对象 res.groups, 该对象实际就是数据库的数据表 res_groups)，该条记录的 name 字段为“Product / Manager”。其中，id="group_product_manager" 是本条记录的标志，在模块范围内必须唯一。

```
<record id="group_product_manager" model="res.groups">
  <field name="name">Product / Manager</field>
</record>
```

product/security 目录下的文件 ir.model.access.csv 有如下行：

```
"id","name","model_id:id","group_id:id","perm_read","perm_write","perm_create","perm_unlink"
"access_product_product_user","product.product.user","model_product_product","base.group_user",1,0,0,0
"access_product_product_manager","product.product
manager","model_product_product","product.group_product_manager",1,1,1,1
```

该两行定义了组 base.group_user、product.group_product_manager 对对象 model_product_product 的读、写、建、删的权限。Product.group_product_manager 表示 product 模块中的 id=group_product_manager 的组，这正是“Product / Manager”。“Product / Manager”组对产品对象有完全的权限。base.group_user 组在 base 模块中定义，是系统的普通用户。从该权限定义知道，所有用户都允许查看产品，但只能读，不能增删改。

又如 sale 模块下的 sale_security.xml 中有下述语句：

```
<record model='ir.ui.menu' id='menu_shop_configuration'>
  <field name="groups_id" eval="[(6,0,[ref('group_sale_manager')])]" />
</record>
```

这个语句会在数据表 ir_ui_menu 中导入一条记录，意思是 id=group_sale_manager 的组，即销售经理组，对 id=menu_shop_configuration 的菜单有访问权限。id=menu_shop_configuration 的菜单在 sale 模块的视图文件 sale_view.xml 中定义，是菜单“设置”(Configuration)。

二、 常见预设权限组

- **Employee:** base 模块定义的权限组，该组拥有业务伙伴、产品、财务、仓库等一些基本对象的读权限，用户一般应该属于该组，否则几乎不能查看和使用系统的任何功能。
- **Administrator / Access Rights:** base 模块定义的权限组，该组拥有创建用户、创建权限组、设置安全规则等权限。具体来说，该组拥有菜单 Administration/Users 和 Administration/Security 的访问权限。
- **Administrator / Configuration:** base 模块定义的权限组，该组拥有系统管理功能，具体来说，该组拥有菜单 Administration/Customization、Administration/Translations、Administration/Modules Management 的访问权限。
- **Useability / No One:** base 模块定义的权限组，该组相当于一个空组。默认情况下，该组没有任何菜单和对象的访问权限，也没有任何 user 属于该组。我能想到的该组的一个用处是，菜单的访问权限，系统默认是，如果没有定义任何组访问该菜单，则任何组都允许访问该菜单。也许某些菜单，希望默认是，如果没有显示声明，任何组都不允许访问该菜单。这种情况，就可以声明 No One 组允许访问该菜单，那么，除非显示声明，否则其他组都不允许访问该菜单了。
- **Useability / Extended View:** 系统的某些扩展功能，如果要使用那些扩展功能，就必须属于该组。一般用户都不必要该组。
- **Partner Manager:** 该组拥有业务伙伴配置和业务伙伴新建功能。默认情况下，只要是 Employee，就能查看业务伙伴信息，但只有 Partner Manager 才能配置、修改、新建、删除业务伙伴信息。
- **Product / Manager:** product 模块定义的权限组，该组拥有产品配置功能，也就是菜单 Products/Configuration 的访问权限。默认情况下，只要是 Employee，就能查看产品信息，但只有 Product / Manager 才能配置和修改产品信息。
- **Finance / Accountant:** 相当于普通会计师，可以进行日常的会计处理工作。
- **Finance / Invoice:** 拥有业务伙伴发票的处理权限，相当于负责收付钱和开票的出纳。也就是对菜单 Financial Management/Invoices 的访问权限。
- **Finance / Manager:** 相当于财务管理员，可以配置财务模块，以及查看财务报表。
- **Sale / Manager:** 允许配置销售模块，也就是允许访问菜单 Sales Management/Configuration。
- **Sale / Salesman:** 允许进行大部分的销售日常工作，相当于销售部业务员。

第四节 Domain 条件表达式

一、 Domain 条件表达式的应用场合

菜单：合作伙伴 → 合作伙伴、合作伙伴 → 合作伙伴 → 客户、合作伙伴 → 合作伙伴 → 供应商，这三个菜单功能，它们访问的都是合作伙伴对象(res.partner)，显示的格式，即视图也是一样。不同的是，数据过滤条件不同，前者显示所有合作伙伴，“客户”显示分类为客户的合作伙伴，后者显示分类为供应商的合作伙伴。象这样，对象、视图都相同，只是数据过滤条件不同的菜单功能，其过滤条件在 Action（动作）中设定。Action 中设定数据过滤条件，是 Domain 条件表达式的第一个应用场合，也是最普遍的应用。

如下，菜单：管理员 → 低级对象 → 操作 → 窗口对象，检索“供应商”动作，打开“供应商”Action。在这个 Action 定义画面上，可看到“过滤条件”及“环境变量”的字段。Action 的过滤条件表示，本 Action 只显示符合本过滤条件的数据记录，其格式就是 Domain 条件表达式的格式。这里的环境变量值可以理解为全局变量，其中设置的变量可供“过滤条件”表达式使用。如本例，过滤条件也可写作：

```
[('supplier', '<>', default_customer)]
```

这等同于

```
[('supplier', '<>', 0)]
```

过滤条件:	[('supplier', '=', 1)]
环境变量值:	{ 'default_customer': 0 }
最大显示数:	80
视图模式:	tree, form

图 十-6 Action 的过滤条件

Domain 表达式的第二个最普遍的应用场合是，例如，在合作伙伴表单上，点击“国家”字段右边的查询图标，会弹出如下的国家查询画面：

查找 国家

基本查询 高级查询

? 国名: ? 国家代码:

过滤 新建 选择

关闭

<< 第一页 < 上一页 [1 - 20 共 80] 下一页 > 最后一页 >>

国名	国家代码
<input type="radio"/> 不丹	BT
<input type="radio"/> 东帝汶	TP
<input type="radio"/> 东萨摩亚	AS
<input type="radio"/> 中国	CN
<input type="radio"/> 中国台湾省	TW

图 十-7 字段查询的过滤条件

在字段的查询画面中，如果希望只显示符合特定条件的记录，则可以在视图中该字段定义处，加上过滤条件（domain）。字段定义画面参看第二章说明。

Domain 表达式的其他一些应用，包括，如前一节的组的规则设定中，隶属表达式也是 Domain 的应用场合。在代码编写中，对象的 search 方法也可带有 Domain 表达式检索，如下例：

```
obj = self.pool.get('res.partner')
ids = obj.search(cr, uid, [('user_id','=',uid)])
```

这个例子，首先从对象池中取得合作伙伴对象，再调用对象的 search 方法。该方法带有过滤条件：只检索当前用户对应的合作伙伴。

二、 Domain 条件表达式写法

最简单的格式如下：

```
[('字段', '操作符', 值)]
```

例如：

```
[('active', '=', True)]
```



```
[('id', 'in', [7, 13, 42, 666])]
```

[('active', '=', True)] 等同于 SQL 条件: `where active = true`

[('id', 'in', [7, 13, 42, 666])] 等同于 SQL 条件: `where id in (7, 13, 42, 666)`

Domain 条件表达式中的'字段'必须是对象的字段名, 可用操作符有: `=, <>, <=, >=, in, child_of`。值必须是一个常量值或环境变量 (context) 中的变量值。环境变量 (context) 可以自己设置, 如前述 Action 定义画面中的“环境变量值”字段。

Domain 条件表达式也可用关系操作符串起多个简单条件式。如下例:

```
[('field', '=', 'value'), ('field', '<>', 42)]
```

其含义是: (`field = 'value' AND field <> 42`), 缺省关系操作符是 AND.

```
[('&', ('field', '<', 'value'), ('field', '>', 'value'))]
```

其含义是: (`field < 'value' AND field > 'value'`)

更复杂的例子如下:

```
[('|', ('field', '=', 'value'), ('field', '=', 'value'))]
```

(`field = 'value' OR field = 'value'`)

```
[('&', ('field1', '=', 'value'), ('field2', '=', 'value'), ('|', ('field3', '<>', 'value'), ('field4', '=', 'value')))]
```

(`field1 = 'value' AND field2 = 'value' AND (field3 <> 'value' OR field4 = 'value')`)

```
[('&', ('|', ('a', '=', 1), ('b', '=', 2)), ('|', ('c', '=', 3), ('d', '=', 4)))]
```

((`a = 1 OR b = 2`) AND (`c = 3 OR d = 4`))

```
[('|', (('a', '=', 1), ('b', '=', 2)), (('c', '=', 3), ('d', '=', 4)))]
```

((`a = 1 AND b = 2`) OR (`c = 3 AND d = 4`))

第十一章 OpenERP 翻译方法

OpenERP 提供了很方便的国际化支持技术和工具。本章详细介绍了 OpenERP 用户界面翻译的三种方法：资源术语翻译，通用术语翻译，批量翻译或纠正翻译。而后浅析了 OpenERP 国际化技术的实现原理。三种翻译方法中，“资源术语翻译”法最方便，最简单，但缺点是不是所有的文字都可以用该方法翻译，大约有 20%的文字该方法不能翻译。“通用术语翻译”法能翻译所有文字，方法也简单，但操作比较繁琐，适合于个别术语的翻译，不适合大批量翻译。第三种方法能翻译所有术语，也适合大批量翻译，但方法稍复杂，适合于开发人员新模块汉化时使用。

第一节 资源术语翻译

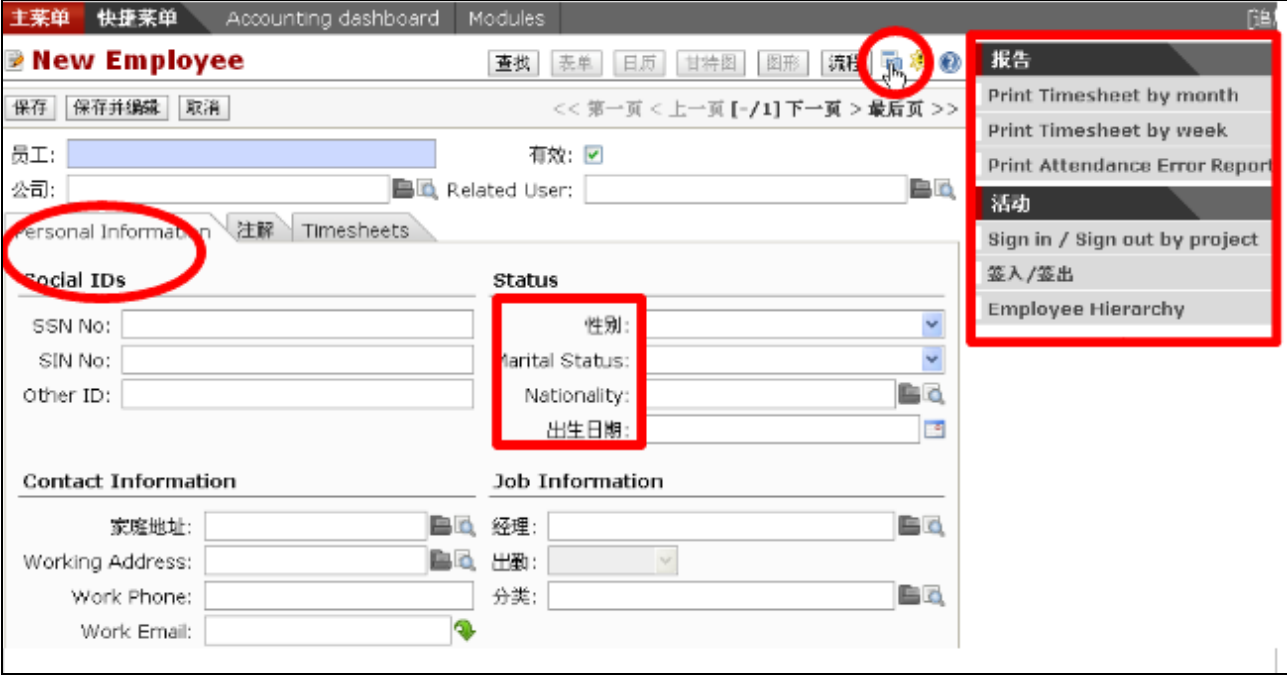
在 openERP 中，一个典型的业务对象，如“合作伙伴”，“员工”，“地址”等，有四部分需要翻译。一是对象的属性字段，二是对象属性字段在屏幕视图上的显示标签，三是对象关联的视图上其他文字，四是画面右边的对象关联的操作。当以表单形式查看 Open ERP 对象时，画面右上方有个“翻译资源”的按钮，点击该按钮，进入翻译画面。在翻译画面，可以分别翻译该业务对象的字段、标签、视图、关联操作四部分的术语。翻译好以后，刷新画面立即可以看到翻译效果，无需重启服务器。

下面介绍该方法的详细操作步骤。

1) 选择业务对象。如下图，点击“New Employee”则进入员工业务对象画面。



2) 进入业务对象画面，点击右上角的“翻译资源”按钮（下图右上角红色小圆圈所示），则进入翻译画面。下图左边椭圆所示为对象关联视图上的术语，中间方框所示为对象属性字段在视图上的显示标签，右边方框所示为对象关联的操作。



3) 翻译画面上逐个翻译术语并保存。下拉列表中字段、标签、关联、视图分别对应业务对象的属性字段，属性字段显示标签，对象关联操作、视图上标签以外的文字。



4) 翻译好以后，刷新业务对象画面，立即可以看到翻译效果。

注：属性字段、标签、关联操作，无需重启服务器，立即能看到翻译效果，但视图上的其他文字，目前版本好像还要重启服务器才能看到效果。

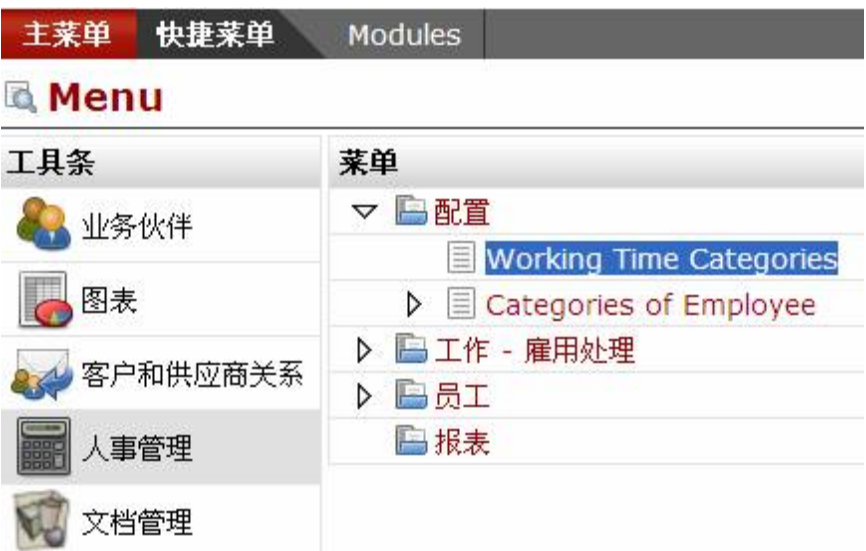


第二节 通用术语翻译

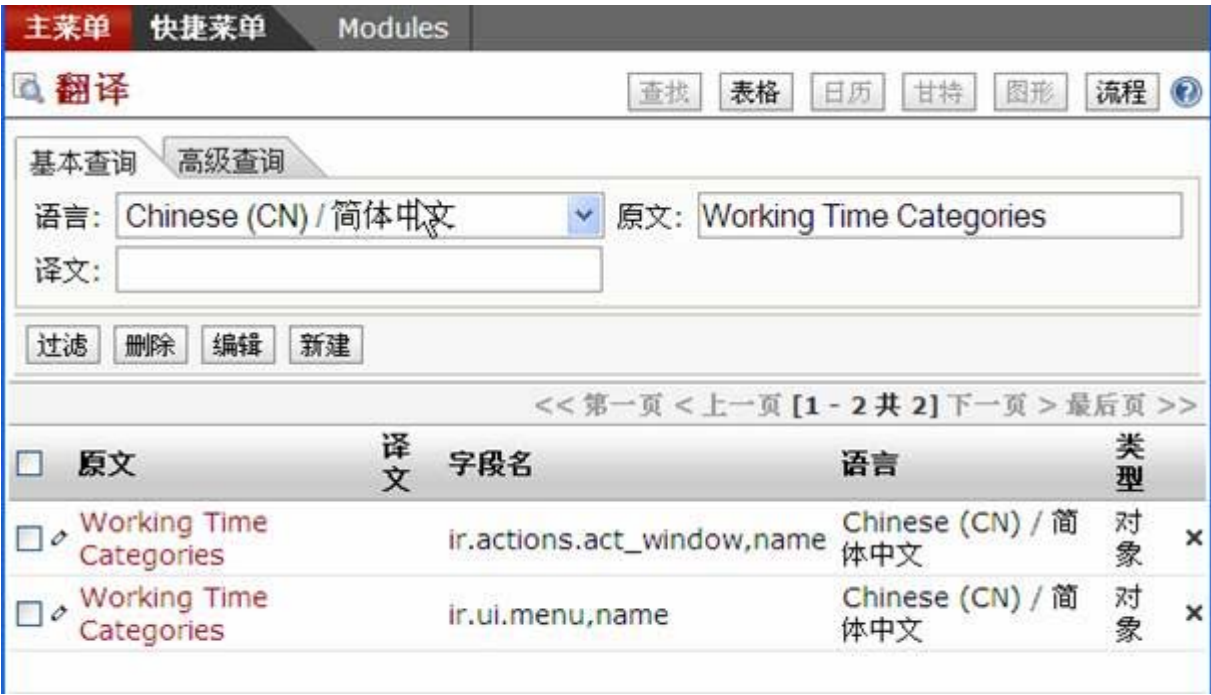
在任何画面上，看到没翻译的术语，选择之，进入菜单“管理员 → 翻译 → 术语 → 所有术语 → 未翻译术语”。语言选择“Chinese(CN)简体中文”，原文中输入待翻译术语(英文)，点击“过滤”按钮，该术语出现过的记录都会显示出来，逐条翻译之。要看到翻译效果，必须重起服务器。

下面介绍详细操作步骤。

1) 选择待翻译词条



2) 查找词条，点击过滤按钮查找



3) 点击词条，进入翻译画面，点击编辑按钮，翻译词条并保存



第三节 批量翻译或纠正翻译

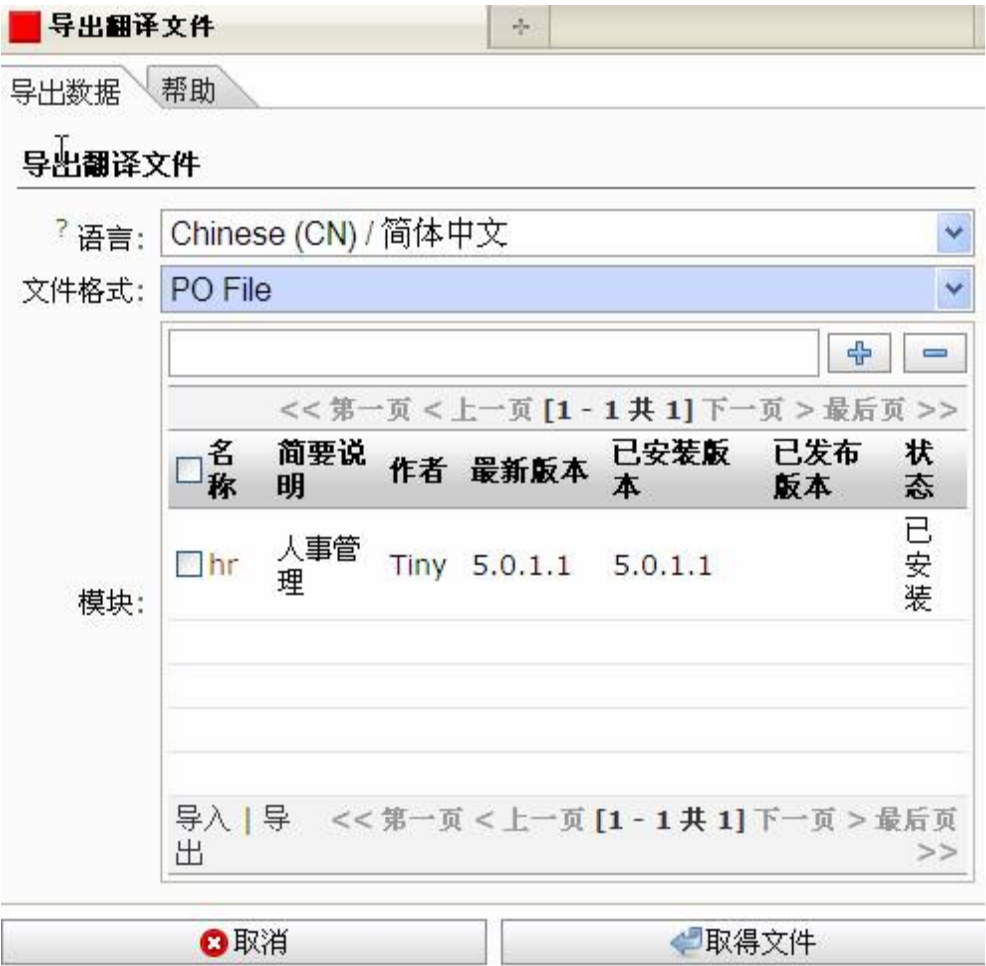
当翻译量很大时，逐个翻译词条比较麻烦。当发现有不恰当的翻译时，因为已经翻译过，在“未翻译术语”菜单中查找不到，不能用通用术语方法修正，这时就要用此处的方法处理。进入菜单“管理员->翻译->导入/导出->导出翻译文件”。语言选择“Chinese(CN)简体中文”，文件格式选择“CSV”或“PO”，点击按钮“+”选择要导出的模块，点击“取得文件”按钮导出文件。在文本编辑器中翻译导出的文件。

译好以后，进入菜单“管理员 → 翻译 → 导入/导出 → 导入翻译文件”。语言名任意填，代码中输入“zh_CN”，点击按钮“更改”，上传翻译好的文件（CSV 或 PO）。重起服务器以后，可以看到翻译效果。

下面介绍详细操作步骤。

1) 导出翻译文件

语言栏不选择任何语言时，导出原始的未翻译文件，选择“Chinese(CN)/简体中文”时，导出带中文翻译的文件，点击按钮“+”时，弹出模块选择按钮选择要导出的模块，不选择任何模块时，将导出所有模块的翻译。



2) 翻译导出的文件

导出文件有.PO 和.CSV 两种格式，两种格式文件都可以用文本编辑器直接打开翻译。文件翻译好以后，注意保存为 utf-8 编码格式，不要存成 gb2312 格式。文本编辑软件建议用 EmEditor,该软件对亚洲语言支持很好。

PO 文件格式，msgid 是英文，msgstr 是对应的中文翻译。

```
# Translation of OpenERP Server.
# This file contains the translation of the following modules:
#
#   * account
#
msgid ""
msgstr ""
"Project-Id-Version: OpenERP Server 5.0.1\n"
"Report-Msgid-Bugs-To: support@openerp.com\n"
"POT-Creation-Date: 2009-08-06 15:12:04+0000\n"
```

```
"PO-Revision-Date: 2009-08-06 15:12:04+0000\n"
```

```
"Last-Translator: <>\n"
```

```
"Language-Team: \n"
```

```
"MIME-Version: 1.0\n"
```

```
"Content-Type: text/plain; charset=UTF-8\n"
```

```
"Content-Transfer-Encoding: \n"
```

```
"Plural-Forms: \n"
```

```
#. module: account
```

```
#: field:account.tax.template,description:0
```

```
msgid "Internal Name"
```

```
msgstr "内部名称"
```

```
#. module: account
```

```
#: view:account.tax.code:0
```

```
msgid "Account Tax Code"
```

```
msgstr ""
```

```
#. module: account
```

```
#: model:ir.actions.act_window,name:account.action_invoice_tree9
```

```
msgid "Unpaid Supplier Invoices"
```

```
msgstr "未支付供应商票据"
```

```
#. module: account
```

```
#: model:ir.ui.menu,name:account.menu_action_invoice_tree9
```

```
msgid "Unpaid Supplier Invoices"
```

```
msgstr "未支付供应商票据"
```

CSV 文件格式，倒数第二栏是英文，最后一栏是对应的中文翻译。

```
module,type,name,res_id,src,value
```

```
account,wizard_field,"account.general.ledger.report,checktype,date_from",0, Start date,开始日期
```

```
account,wizard_field,"account.partner.balance.report,init,date1",0, Start date,开始日期
```

```
account,wizard_field,"account.third_party_ledger.report,init,date1",0, Start date,开始日期
```

```
account,wizard_field,"account.third_party_ledger.report,init,reconcil",0, Include Reconciled Entries,包括  
调节分录
```

```
account,wizard_view,"account.fiscalyear.close.state,init",0, Close states of Fiscal year and periods,
```

```
account,field,"wizard.multi.charts.accounts,code_digits",0,# of Digits,数字
```

3 导入翻译文件

导入语言

取消

Ok

导入新语言

语言名:

代码 (例: zh_CN):

文件:

浏览

请导入UTF-8编码的.CSV文件, 如果是.csv文件, 请检查文件第一行是不是下述之一:

type,name,res_id,src,value

module,type,name,res_id,src,value

您也可以导入.po文件

第四节 OpenERP 国际化原理浅析

OpenERP 的每个模块目录下, 有个 i18n 的目录, 这个目录下放的就是本模块的语言包。例如, 简体中文语言包的文件名是”zh_CN.po”, 繁体中文语言包是”zh_TW.po”。安装或更新模块时, 根据你创建账套时的语言选项, 对应的语言包 PO 文件会导入到数据库表”ir_translation”。如下图是简体中文导入 ir_translation 表后的数据例子。

id	lang	src	name	type	res_id	value
[PK]	character	text	character varying(128)	character va	integer	text
1	zh_CN	SMS - Gateway: clickatell	res.partner.sms_send,init	wizard_view	0	SMS-网关: clickatell
2	zh_CN	No journal for ending writing has	addons/addons/account/wizard/wizai	code	0	"
3	zh_CN	%j - Day of the year as a decimal	res.lang	view	0	%j - 年度天数 [001,366]。
4	zh_CN	Product quantity	addons/addons/mrp/report/price.py	code	0	产品数量
5	zh_CN	Metadata	ir.values,meta_unpickle	field	0	元数据
6	zh_CN	View Architecture	ir.ui.view,arch	field	0	视图结构
7	zh_CN	View Architecture	ir.ui.view.custom,arch	field	0	视图结构
8	zh_CN	You can not create this kind of d	addons/addons/base/ir/ir_model.py	code	0	您没有权限创建该类型文档!
9	zh_CN	Code (eg:en_US)	module.lang.import,init,code	wizard_field	0	代码 (例: zh_CN)
10	zh_CN	Workflow	workflow	view	0	工作流
11	zh_CN	Workflow	workflow.activity,wkf_id	field	0	工作流
12	zh_CN	Workflow	workflow.instance,wkf_id	field	0	工作流
13	zh_CN	Account move line "%s" is not va	addons/addons/account/account_ba	code	0	"
14	zh_CN	Hungarian / Magyar	module.lang.install,init,lang	selection	0	"
15	zh_CN	Workflow On	ir.actions.server,wkf_model_id	field	0	"
16	zh_CN	The Bank type %s of the bank ac	addons/addons/l10n_ch/wizard/dta_	code	0	不支持银行账号: %s 所属的
17	zh_CN	Created Views	ir.module.module	view	0	创建视图
18	zh_CN	You can not delete posted move	addons/addons/account/account.py	code	0	"
19	zh_CN	Outgoing transitions	workflow.activity	view	0	出迁移
20	zh_CN	Yearly	ir.report.custom,frequency	selection	0	每年
21	zh_CN	The unlink method is not impleme	addons/osv/orp.py	code	0	"
22	zh_CN	Target Window	ir.actions.act_window,target	field	0	目标窗口
23	zh_CN	Choose between the "Simplified I	ir.actions.todo,note	model	2	选择“简单界面”或“扩展界面

该表中, 各字段含义如下:

lang: 语言种类, zh_CN 表明是简体中文。

src: 原始文字，即要翻译的文字，通常是英文。

value: 翻译后的文字。

name, type, res_id 三个字段根据文字串的类型不同，含义不同。下面举例说明：

type=code: 这表示本文字串是硬编码在代码行中的字符串，如错误提示文字多属于此类。**name** 字段是源代码的文件名，**res_id** 无意义。OpenERP 运行时，碰到代码中的需要翻译的字符串时，通过条件：**lang=语言种类 and type=code and name=源文件名 and src=待翻译字符串** 查找本表，如果 **value** 不为空，则界面上显示 **value** 值，否则显示原始文字串（英文）。代码中，待翻译字符串以下划线及括号标明，如上图中的第 4 行的 **Product quantity**，在源代码的出现形式是：**_('Product quantity')**。

type=field: 这表示本文字串是对象字段的标签。**Name** 是对象和字段，**res_id** 无意义。如上图第 5 行，**name=ir_values,meta_unpickle** 表示对象“**ir_values**”的字段“**meta_unpickle**”的标签。

type=model: 这表示本文字串是对象资源的值，最常见的是菜单。**name** 是对象和字段，**res_id** 是资源 id。如 **name= ir.ui.menu,name res_id=58** 表示的是对象“**ir.ui.menu**”中的 **id=58** 的菜单项的显示名。

type=view: 这表示本文字串出现于视图中。**name** 是该视图对应的对象名，**res_id** 无意义。如上图第 3 行表示，对象 **res.lang** 的视图中的字符串“**%y - Year without century as a decimal number [00,99].**”。

还有其他一些类型，这里就不一一列举了。OpenERP 运行时，碰到要翻译的字符串时，根据该字符串的类型，分别采用不同的查找方法，从 **ir_translation** 表查得翻译值 **value**，显示在画面上。

第十二章 OpenERP 模块开发

一个真实的请假流程，通常要考虑如下要求：

- R1：普通员工只可以看见自己的请假单。
- R2：只有主管才可以批准请假单。
- R3：主管不应该看见草稿状态的请假单。
- R4：主管只可以看见自己部门的请假单。
- R5：主管可以查询自己部门的非草稿状态的所有请假单。
- R6：请假申请时发 Mail 通知主管，有待审批请假单。

本章继续前面章节的例子，开发一个完整的请假模块，实现上述要求。

第一节 OpenERP 模块结构

本节先安装示例代码，演示代码功能，而后介绍示例代码结构，解释 OpenERP 模块的组成部分及各部分的作用。

一、安装示例代码

将本章附件代码中的”qingjia”目录拷贝到 OpenERP 的安装目录下的 addons 下，点击菜单：管理员 → 模块管理 → Update Modules List 。检查模块更新后，在未安装模块下会出现新模块”qingjia”，安装它。请假模块在系统中增加菜单：

请假申请
 请假单审批
 所有请假单
 我的请假单

其中：

- 我的请假单：显示当前用户的所有请假单，包括未提交、被拒的、已批的、待批的所有请假单。所有员工都可以看见该菜单。
- 请假单审批：显示本部门的待批请假单，只有部门经理才能看见该菜单，普通员工看不见该菜单。
- 所有请假单：显示本部门的所有已提交的请假单，包括待批的、被拒的、已批的。只有部门经理可以看见该菜单。

请假模块还增加权限组 “qingjia / department manager”， workflow 角色 “部门经理”，以及下述用户：

登录名	登录密码	用户名	主管	权限组	角色
yg01	123	员工 1	经理 1		
yg02	123	员工 2	经理 2		
jl01	123	经理 1		department manager	部门经理
jl02	123	经理 2		department manager	部门经理

以员工身份登录系统，只能看见“我的请假单”菜单。以经理身份登录系统，三个菜单都能看见，但只能看见自己部门的请假单。员工在“我的请假单”中创建请假单，送批。而后，虽然画面上有“批准”

和“拒绝”按钮，但员工不拥有“部门经理”角色，点击按钮无效，请假单状态不会变化。此处有点小 Bug，据说下一版本中，不拥有相应角色时，将看不见相应按钮。

二、 请假模块结构

请假模块各文件及目录如下：

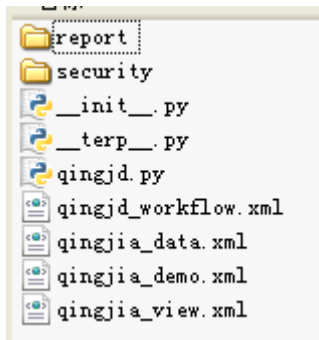


图 十二-1 字段查询的过滤条件

- `__init__.py` : 和普通 Python 模块中的 `__init__.py` 作用相同，是每个 OpenERP 模块必须的。
- `__terp__.py` : OpenERP 模块特有的，详细内容见后文，是每个 OpenERP 模块必须的。
- `qingjd.py` : 请假单对象定义代码。
- `qingjia_view.xml` : 请假单视图（列表和表单视图）、动作（Action）、菜单定义文件。
- `qingjd_workflow.xml` : 请假 workflow 定义文件。
- `qingjia_data.xml` : 模块基础数据定义文件，本例中的员工和经理信息定义在该文件中。基础数据是模块正常运行必须的数据。
- `qingjia_demo.xml` : 演示数据定义文件，本例实际未使用该文件。演示数据不是模块必须的数据。
- `security` : 权限设置文件。通常包括两部分，一是定义新的权限组(group)和角色，二是定义权限组对对象(Object)的访问权限（读写建删）。
- `report` : 报表定义相关的文件。本例的“打印请假单”报表在此定义。
- `process` : 企业流程（Enterprise Process）定义相关文件，本例未使用。

上述各个部分，只有 `__init__.py` 和 `__terp__.py` 是必须的，其他的都不是必须的。

本例 `__terp__.py` 文件内容如下：

```
"name": "Employee Leave Example",
"author": "NewZN",
"version": "0.1",
"depends": ["base", "process", "hr"],
"description": """《OpenERP 应用和开发基础》代码模块开发示范例子。
""",
"init_xml": [],
"update_xml": [
    'security/qingjia_security.xml',
    'security/ir.model.access.csv',
    'qingjia_view.xml',
    'qingjd_workflow.xml',
    'report/rpt_qingjd.xml',
```

```
'qingjia_data.xml',
],
"demo_xml": [],
"installable": True,
```

其中：

- **name** : 模块名，任意文字。
- **depends** : 依赖模块，即安装本模块时将检查此处定义的模块，如果没安装，将自动一起安装。通常所有模块都要依赖 **base** 模块。本例工作流要依赖 **process** 模块，员工及部门经理关系用到 **hr** 模块。
- **description** : 模块描述，任意文字。
- **init_xml** : 当 Server 以参数 `- init=module` 启动时要加载的 XML 数据文件。格式是“路径+文件名”，路径是相对模块目录的路径。
- **update_xml** : 模块安装和升级时需要重新加载的 XML 文件，视图、报表、权限、工作流等的定义文件通常放在此处。通常权限定义文件放在前面，因为其它文件常引用权限定义数据。
- **demo_xml** : 演示数据文件。
- **installable** : 通常固定为 `True`。

第二节 理解 OpenERP 的代码

OpenERP 的代码实际上是以 Python 及 XML 格式定义对象、视图、动作（Action）、菜单、工作流、报表、企业流程、权限组、角色等。这些基本概念前述章节已经介绍过了，本节主要介绍和前述章节不同之处。

一、对象定义

本例的请假单对象定义代码如下：

```
# -*- encoding: utf-8 -*-
from osv import osv, fields
class qingjia_qingjd(osv.osv):
    _name = 'qingjia.qingjd'
    _description = '请假单'
    def get_employee(self, cr, uid, context={}):
        obj = self.pool.get('hr.employee')
        ids = obj.search(cr, uid, [('user_id','=',uid)])
        res = obj.read(cr, uid, ids, ['id','name'], context)
        return res and res[0]['id'] or 0
    _columns = {
        'shenqr': fields.many2one('hr.employee', '申请人', required=True),
        'tians': fields.float('请假天数', required=True),
        'kaisrq': fields.date('开始日期', required=True),
        'shiyous': fields.text('请假事由'),
        'active': fields.boolean('有效'),
        'state': fields.selection([('draft','草稿'),('wait_prove','待批'),('proved','已批'),('rejected','被拒')], '状态',
required=True)
    }
    _defaults = {
        'shenqr': lambda self,cr,uid,context: self.get_employee(cr,uid,context),
        'active': lambda *a: 1,
```

```
'state': lambda *a: 'draft',
}
qingjia_qingjd()
```

请假单对象和之前章节中界面上定义的对象基本相同。值得注意的有：

- 1) 第一行#-*- encoding: utf-8 -*-，这行指出文件编码格式为 utf-8。当代码行中有汉字时，必须有本行，而且，文件必须以 utf-8 格式保存。通常，多数编辑器的缺省保存格式是 gb2312，要注意手工改动保存格式。
- 2) 对象名和字段名没有“必须以 x_开头”的限制。
- 3) 申请人字段，和前面章节不同的是，此处是到 hr.employee 的 many2one 关系。主要是因为权限设置时要用到主管等部门信息，这些信息在 employee 中，res.users 中没有。另外，此处设定申请人缺省值是当前登录用户对应的 employee。该对应的 employee 通过函数 get_employee 取得。
- 4) Active 字段，OpenERP 的多数对象带有该字段。该字段的作用是，如果 Active 为 False，OpenERP 将自动过滤掉本记录。这相当于“软删除”记录，即数据库中还有，但任何界面上都找不到该条记录。
- 5) State 字段，设置了该字段的缺省值为“draft”('state': lambda *a: 'draft')。lambda 是 Python 的语法，相当于函数的简写格式。'state': lambda *a: 'draft' 等同于下述方法定义：

```
def xfun(self, cr, uid, context):
```

```
    return 'draft'
```

```
state=self.xfun( cr, uid, context)
```

- 6) 最后一行的 qingjia_qingjd()，表示创建请假单对象，并将请假单对象放入对象池中。以后只要用语句 self.pool.get('qingjia.qingjd') 即可取得该对象。通过该对象即可对相应数据表进行增、删、查、改等数据库操作。

- 7) 对象定义的完整属性如下：

必须属性

 _name 对象名

 _columns 对象字段定义

可选属性

 _table 对象对应的表名

 _description

 _defaults

 _order

 _rec_name name 字段对应的数据库字段名

 _auto 是否自动创建数据库表

 _constraints

 _sql_constraints

 _inherit

 _inherits

上述属性的详细含义请参考官方文档。

- 8) OpenERP 支持的字段类型如下：

基础类型

 char, text

 boolean, integer, float

 date, time, datetime

 binary

复杂类型

 selection, function, related

关系类型

```

one2one
one2many
many2one
many2many

```

上述类型的详细用法请参考官方文档。

二、视图、动作及菜单定义

视图、动作和菜单定义，参看文件 `qingjia_view.xml`。视图定义和前面章节基本相同，不同之处是，申请人和状态字段设为只读。也就是限定，状态不可任意改变，申请人只可以为自己。下面重点看看请假单审批菜单。

```

<record model="ir.actions.act_window" id="action_qingjia_qingjdprove">
  <field name="name">请假单审批</field>
  <field name="res_model">qingjia.qingjd</field>
  <field name="domain">[(('state','=','wait_prove'),('shenqr.parent_id.user_id','=','uid'))]</field>
</record>
<menuitem name="请假单审批" parent="menu_qingjia_main"
  id="menu_qingjia_qingjdprove" action="action_qingjia_qingjdprove"/>

```

值得注意的是：

1) OpenERP 的 XML 的数据定义格式。其数据定义的通用格式如下：

```

<record model="..." id="...">
  <field name="...">Value</field>
  .....
</record>

```

`<record>` 定义在数据表中增加一条记录。其中，`model` 是对象名，也等同于数据表名。`id` 是任意值，但 `id` 中的“.”有特殊含义，它是模块分界符。`id` 标志本条数据，在本模块范围内必须唯一。引用其它模块的数据时，必须用“`module.name.id`”引用。`field` 定义字段值，其“`name=`”指明字段名，`Value` 是字段值。

本例的 `<record>` 在数据表 “`ir_actions_act_window`” 中插入一条记录 (`name=`‘请假单审批’，`ren_model=`‘qingjia.qingjd’，`domain=`...)。

2) 本 Action 通过 Domain 条件表达式 “`[(('state','=','wait_prove'),('shenqr.parent_id.user_id','=','uid'))]`”，限定本 Action 仅显示状态为“待审批”的，且申请人的主管(`shenqr.parent_id.user_id`)是当前登录用户(`uid`)的请假单。

3) `menuitem` 中的 `action="action_qingjia_qingjdprove"` 定义本菜单关联的 Action。

4) 可以指定每个 Action 的视图。每个 Action 可以同时有 “`tree, form, calendar, graph, gantt, process`” 六个视图，它们分别对应 Web 界面右上方的六个按钮。最常见的是 `tree`（列表）和 `form`（表单）视图。如果不指定 Action 的视图，系统自动采用 Action 中的对象 (`res_model`) 关联的默认视图，本例是“请假单”对象的默认列表和表单视图。

```

<record id="stock.action_picking_tree6_view1" model="ir.actions.act_window.view">
  <field eval="1" name="sequence"/>
  <field name="view_mode">tree</field>
  <field name="view_id" ref="stock.vpicktree"/>
  <field name="act_window_id" ref="stock.action_picking_tree6"/>
</record>

```

上例定义 Action “`stock.action_picking_tree6`”（库存模块中的 Action: `action_picking_tree6`）的列表视图为 “`stock.vpicktree`”（库存模块中的视图: `vpicktree`）。`view_mode` 指定视图类型，即 “`tree, form, calendar, graph, gantt, process`” 之一。`Sequence` 指定视图的显示顺序，数字越小越先显示，通常 `tree` 总是最先显示。

三、 workflows和报表定义

工作流定义文件参看 qingjd_workflow.xml, 和前面章节不同的是增加了工作流角色:

```
<field name="role_id" ref="role_depmanager" />
```

角色 "role_depmanager" 在文件 "security/qingjia_security.xml" 中定义。此处 ref 表示引用该条角色定义记录。在 qingjia_data.xml 中, 定义经理时, 下述语句赋予经理 "部门经理" 角色。

```
<field name="roles_id" eval="[(6,0,[ref('role_depmanager')])]" />
```

报表定义参看 report 目录下的相关文件。报表和前述章节完全一致, 没有改变。

四、 权限定义

文件 security/ir.model.access.csv 定义权限组对对象的访问权限。

```
"id","name","model_id:id","group_id:id","perm_read","perm_write","perm_create","perm_unlink"
"access_qingjia_qingjd_depman","qingjia_qingjd_depman","model_qingjia_qingjd","qingjia.group_depmanager",1,1,1,1
```

id 和 name 字段任意取值, model 字段格式为: model_modulename_objectname, 如请假单对象是 "model_qingjia_qingjd"。"qingjia.group_depmanager" 表示 qingjia 模块中的组 "group_depmanager", group_depmanager 是权限组 "qingjia / department manager", 在 security/qingjia_security.xml 中定义。

文件 security/qingjia_security.xml 定义组, 角色, 及组对菜单的访问权限。

第三节 服务端动作(Server Action)

Server Actions 的作用是, 当特定工作流迁移到特定状态时, 触发 Server Action 执行更多动作。如:

- 订单审批通过时, 自动在订单对应的客户中记录一些信息。
- 发票确认时, 自动 Mail 通知有关人员。
- 当一个 CRM 案例结束时, 自动弹出一个窗口要求填入总结信息。

本节通过服务端动作(Server Action)实现请假单送批时自动发 Mail 通知主管经理的要求。

一、 配置 Server Action

菜单: 管理员 → 自定义 → Action → Server Action, 新建一个 Server Action, 这个 Server Action 将在请假申请人按下请假 "送批" 按钮时发 Email 通知主管审批。本 Server Action 的各字段说明如下:

- Action Name: 任意取名。
- Object: 本 Action 要操作的对象, 即本 Action 将读取或修改该对象的数据。
- Action Type: Server Action 的类别。OpenERP 提供了多种 Server Action, 本例选用 Email, 其他类型后文再介绍。
- Sequence: 当对应同一动作, 定义了多个 Server Action 时, 本 Action 的执行顺序。数字越小越先执行。
- Condition: 本 Action 是否执行的检查条件。

这几个字段是所有类型的 Server Action 的共同字段, 下面看看 Email Server Action 的特有字段。

- Email Address: 自然是收件人的 Mail 地址。可以直接写地址, 如 abc@hornerp.com。也可以引用对象中

的数据，如下图所示是请假单申请人的主管的工作 Mail (object.shenqr.parent_id.work_email)。注意，此处的对象引用不要用“[[]]”括起来。

- Subject: 其意自明，不用介绍。可用[[]]引用对象数据。
- Message: Mail 正文，可用[[]]引用对象数据，如下例，Mail 开头是请假申请人的主管的名字 (object.shenqr.parent_id.name)。

Server Action

Search Form Calendar Gantt Graph Process

Save Save & Edit Cancel

<< First < Previous [1/1] Next > Last >>

? Action Name: 请假申请Mail通知 ? Object: 请假单 ? Action Type: Email

? Sequence: 5 ? Condition: True

Email Configuration

Email Configuration

? Email Address: object.shenqr.parent_id.work_email

? Subject: 请假申请审批

? Message: [[object.shenqr.parent_id.name]]:
请审批请假申请，审批链接：
http://localhost:8080/tree/open?model=ir.ui.menu&id=353

Access all the fields related to the current object using expression in double brackets, i.e. [[object.partner_id.name]]

Action Type: ir.actions.server

图 十二-2 字段查询的过滤条件

二、配置 OpenERP Mail

要使 OpenERP 发送 Mail，需要在配置文件中配置下述项目：

```
smtp_port = 25
smtp_server = 202.120.88.78
smtp_user = xxx@cc.ecnu.edu.cn
smtp_password = yyy
email_from = xxx@cc.ecnu.edu.cn
```

- smtp_port: 通常是 25，这是 SMTP 协议的默认端口。
- smtp_server: 你的 Mail 服务器。如果不知道你的 Mail 服务 IP，可用 nslookup 命令查找，如下例查找“cc.ecnu.edu.cn”对应的 Mail 服务器。


```
[root@localhost site-packages]# nslookup -q=mx cc.ecnu.edu.cn
Server:          202.121.80.10
Address:         202.121.80.10#53
```

Non-authoritative answer:

```
cc.ecnu.edu.cn  mail exchanger = 10 webmail.ecnu.edu.cn.
cc.ecnu.edu.cn  mail exchanger = 5 mailsan.ecnu.edu.cn.
```

Authoritative answers can be found from:

```
cc.ecnu.edu.cn  nameserver = liwa.ecnu.edu.cn.
```

- **smtp_user**: 登陆 mail 服务器的用户名，通常是一个有效的 Mail 地址。
- **smtp_password**: 登陆 Mail 服务器的密码。OpenERP 将用这里配置好的 Mail 用户名和密码发送 Mail。
- **email_from**: 显示在系统发出的 Mail 的 From 栏的 Mail 地址。通常和 smtp_user 一样。

三、 测试通知 Mail

开始测试之前，还要先修改下 OpenERP 当前版本(本书用的是 Windows 版的 5.06 All In One 版)的几个 Bug。一是不知何故，在 OpenERP 的安装目录下的 Server 下的 library.zip 中，email\mime 下没有任何东西。（也许是 Tiny 公司故意制造某些麻烦，好让用户找他们培训，呵呵）。修正方法是，用 Winrar 或其他工具打开 library.zip，删除其中的 email 目录。再从 OpenERP 安装目录下的 Web\python25\Lib 下的 email 目录拖进 library.zip。Web\python25\Lib 下的 email 是未编译的源码，它下面的 mime 很齐全，原样拖进去即可。注意，不要用 WinRAR 解压开 library.zip，再替换，再压缩成 library.zip。这样的话，将导致 OpenERP Server 无法启动（原因尚不明）。

第二个问题是，目前版本的 OpenERP，对于中文 Email（Subject 或正文中有汉字的 Mail）会报错。修改方法是，Server\addons\base\ir\ir_actions.py (版本 5.06)文件中的下述几行：

```
Line 484: return str(result) → return tools.ustr(result)
Line 549: str(action.subject) → tools.ustr(action.subject)
Line 550: str(action.message) → tools.ustr(action.message)
```

原文中用 str 编码时不能处理 utf-8 的文字，必须改为 ustr 编码。修改完成后重启 Server，以使修改生效。而后，进入菜单：人力资源 → 员工 → 所有员工，找到“经理 1”，点击编辑，在工作 mail (Work Email) 中添加经理的 Mail 地址。再进入菜单：管理员 → 自定义 → 工作流，选择“请假申请工作流”，打开，编辑 Activity “wait_prove”，在该 Activity 中增加前面配置的 Server Action：请假申请 Mail 通知。如下图所示：

Activity	
Name:	wait_prove
Workflow:	请假申请工作流
Kind:	Function
Server Action:	请假申请Mail通知
Python Action:	write({'state': 'wait_prove'})

图 十二-3 字段查询的过滤条件

这个配置表示，当用户按下请假工作流的“送批”按钮时，迁移到 Activity: ” wait_prove”，触发 Server Action: 请假申请 Mail 通知。该 Action 发送 Mail 给主管。

登出系统，以员工 1 身份(yg01)登陆，创建一张请假单，送批，注意，送批动作有点慢。主管将收到通知 Mail，如下：



图 十二-4 字段查询的过滤条件

如果发送成功，在系统日志中会出现如下日志信息：

INFO:email:Email successfully send to : xyz@gmail.com

如果发送失败，日志信息如下：

ERROR:email:[11]: Last revision No. & ID : Failed to send email to : xyz@gmail.com

四、 更多 Server Action

1. Dummy: A dummy stage does nothing. (God knows why its there..)

2. Client Action: You Can select a form to Open, a wizard to run or a report to launch. Select the report from the list.

3. Iteration: Based on a python loop expression, you can iterate server actions, eg: When a stock inward move occurs and you confirm it, you want each line item to be historised.. you can loop on expression `object.move_lines` and create another server action which is referred to do the historising job.

4. Python Code: you can execute a multiline python code. the returned value is the value of the variable `action = { }`. This makes sense only if you want to pop a specific window(form) specific to the context. IMHO you wont need a return value. Note: The code is executed using the `exec` function of python, which is run in the dictionary namespace with variables: `object,time,cr,uid,ids`

5. Trigger: Any transition of the workflow can be triggered using this. The options you need to set are. Workflow Model: The target object on which you want to trigger the workflow. 'Trigger on' field should have the ID of the target model record. Eg have the ID of invoice if you want to trigger a change in invoice. 'Trigger Name' is the signal you have to use to initiate the transition. The drop down lists all possible triggers. Note:Trigger Name list shows all possible transitions from ther models also, so ensure you select the right trigger. Models are given in bracket.

6. Email: Set an email address, Subject & message. You need to configure the Open ERP building smtpserver for this. Power email, a generic email architecture for Open ERP can be used for emailing as it offers a lot more features and automated emails (no messing with server actions required). Its available at: <http://launchpad.net/poweremail> and downloads at <http://launchpad.net/poweremail/+download>

7. SMS: Choose mobile no. eg: `[[object.shipping_address.mobile]]` and the subject.

8. Create Object: This is used to create a new record in any model, when the server action is triggered. The historise feature discussed can be implemented using this. Field mappings are used to give value for fields in record. Lets try an example of creating events for sale order.

1. Destination: Target field for which you are assigning value: eg. partner_id
2. Type: Formula is evaluated using eval and value is directly applied
3. Value/formula: if value just enter it. For partner id the formula will be object.partner_id (refers to partner_id in sale order). Refer to screen shots for more, note that step two has to be completed for actions to run. These screenshots are here only to help understanding. Also note that there is no need of [[]] in formulas. formulas are directly evaluated.
4. Create id: you can choose the field to which the ID of the newly created record has to be saved. In history this is not necessary, however if you have a m2o relationship and want to keep the ID then this is useful
9. Write Object: Similar to create object except that it is used to modify an existing record identified by the write_id. If you are updating a record you previously created in the create, then the same field for create id can be used for write id.
10. Multi actions: as explained in the context of sequence, you can create multiple server actions and call them one by one in the order of their sequences.

第十三章 超强的继承功能

TODO

第十四章 基于 XML-RPC 的集成

TODO