本书下载地址:

http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1

本书是《OpenERP 应用和开发第三章》,上述坛子是本书的连载基地, 之前和之后的章节请到上述坛子里找。

OpenERP 应用和开发基础 (第三章)

作者: 老肖

msn: saiwai76@hotmail.com

mail: xiangfu.xiao@gmail.com

QQ: 1417063315

Shine-it注册名: NewZN

OpenERP中国本地化项目: http://code.google.com/p/hornerp/

更新日期:

2009年12月24日,完成第一章

2009年12月28日,添加第二章

2009年12月30日,添加第三章

2010年01月20日,第三章中添加企业流程(Enterprise Process) 一节

目录

第一章	简单进销存案例	1
第二章	OpenERP 开发入门	2
第三章	工作流和报表	3
第一节	ウ 理解工作流(Workflow)	3
第二节	5 实作工作流	4
-	-、 修改对象和视图	4
_	二、 创建工作流、活动和迁移	6
Ξ	三、 测试工作流	9
第三节	方 企业流程(Enterprise Process)	9
第四节	芍 OpenERP 报表开发的预备知识	11
_	一、 报表开发工具	11
_	二、 XSL 简介	12
第五节	ち OpenERP 报表简介	13
_	一、 基于 sxw 或 odt 模板的报表	13
_	二、 基于 xsl+xml 模板的报表	15
第六节	· 报表中文化解决方法	15
_	一、 让报表显示汉字	15
_	二、 让报表说中文	17
第七节	5 实作报表	17
_	一、 编写 RML 文件	17
_	二、 编写报表对象代码及 XML 文件	17
Ξ	三、 测试	18
第四章	OpenERP 的权限设置	20
第五章	OpenERP 代码开发	20
第六章	超强的继承功能	20
第七章	基于 XML-RPC 的集成	20

_,

 \equiv

第一章 简单进销存案例

下面是本章的目录,本章内容请到坛子里找,坛子地址:

http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1

第一节 进销存业务流程 1 采购流程1 —, _, 销售流程1 三、 系统功能2 第二节 openERP系统安装2 —, Windows 版安装 3 初次安装者常见问题 三、 中文化包的导入 7 第三节 从源码安装 openERP 8 Linux 版安装 8 —, Windows 版安装 13 第四节 模块配置14 —, 财务设置14

设置业务伙伴及产品信息

确认库位和仓库 20

17

第五节测试采购、销售流程 22一、采购流程 22二、销售流程 28

第二章 OpenERP 开发入门

下面是本章的目录,本章内容请到坛子里找,坛子地址:

http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1

第一节 基本概念 32 第二节 实例演练 34 一、 创建请假单对象 34 二、 创建视图 35

三、 创建菜单和系统动作(Action)37

四、测试39五、视图编辑器40第三节深入数据库41

第三章 工作流和报表

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:

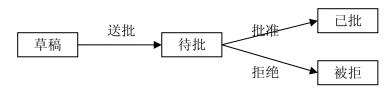


图 三-1 请假单状态图

当我们关注处理步骤时,**活动图**如下:

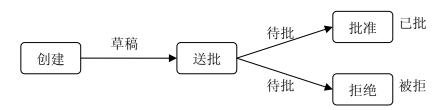


图 三-2 请假申请活动图

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),在上述的请假申请工作流中,有3个迁移。迁移又分**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 \rightarrow 自定义 \rightarrow 数据库结构 \rightarrow 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。

② 字段说明

\capacal \	x_state 状态		
字段类型		属性	
字段类型: Object Relation: Relation Field: Field Selection: 尺寸大小: 手动创建: 领域:	NULL [('draft','草稿'),('	要求: 仅读: 可查询: Translate: [?] 正在删除:	□ 不可查询 ∨

图 三-3 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 \rightarrow 自定义 \rightarrow 用户界面 \rightarrow 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:

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。



图 三-4 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。
- · **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。



活动							
名称	工作流	类别	Duthonx114F "	· 多端动	流程开	流程停	>
⊘ draft	请假申请工 作流	功 功 能	write({'x_state':'draft'})	=	始 是	止 否	×
<pre>ø wait_prove</pre>	请假申请工 作流	功 能	write({'x_state':'wait_prove'})		否	否	×
<i>o</i> proved	请假申请工 作流	功 能	write({'x_state':'proved'})		否	是	×
<i>o</i> rejected	请假申请工 作流	功 能	write({'x_state':'rejected'})		否	是	×

图 三-5 工作流定义画面

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。
- · **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。
- · **分叉模式和合并模式**:请参考官方文档,从略。

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,在这里定义迁移会报错。

通活动

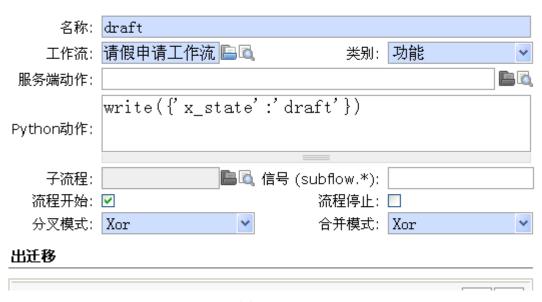


图 三-6 工作流活动定义画面

迁移

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



图 三-7 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
- · 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**: 非按钮触发迁移时,这里指定触发迁移的对象及对象的触发方法。系统会定期检查触发器对象的触发条件式(方法),如果返回 True 则触发迁移。



图 三-8 工作流迁移定义画面

至此,"请假申请工作流"定义完毕,下面测试该工作流。

三、 测试工作流

回到主菜单,点击 请假申请 → 请假单,新建一张请假单,如下图示: 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 三-9 请假申请工作流测试画面

本例的工作流还有一些问题,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 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。

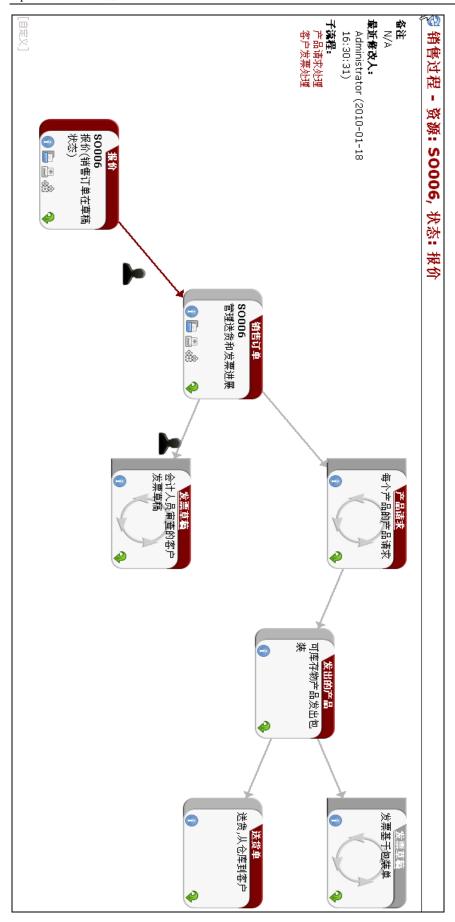


图 三-10 企业流程示意图

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,企业流程只不过是以流程图的方式,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。

第四节 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'</pre>

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>

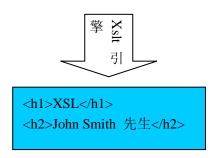


图 三-11 XSLT 样式处理示意图

第五节 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。

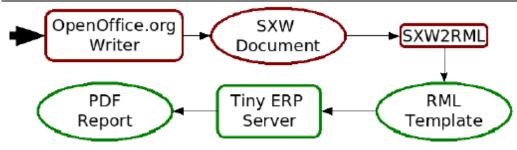


图 三-12 SXW 报表处理过程

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 \rightarrow 模块管理 \rightarrow 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 \rightarrow 低级对象 \rightarrow 操作 \rightarrow 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。

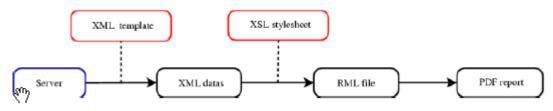


图 三-13 xs1+xml 报表处理过程

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.xsl、 base\res\partner\report\partner_report.xml。

第六节 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,汉字只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.xsl"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。

第七节 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, Qing, ID)]] OpenERP应用和开发基础 - 请假单							
请佣	假人 [[QingJD.x_shenqr.name]]						
请假	民天数	[[QingJD.x_tians]]	开始日期	[[QingJD.x_kaisrq]]			
请假	最事由	[[QingJD.x_shiyou]]					
审批	审批状况 [[QingJD.x_state]]						
申请	日期:	[[QingJD.create_date]]					

图 三-14 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

pt_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_qingid**: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

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 三-15 打印请假单

点击请假单,可看到如下 PDF 报表:

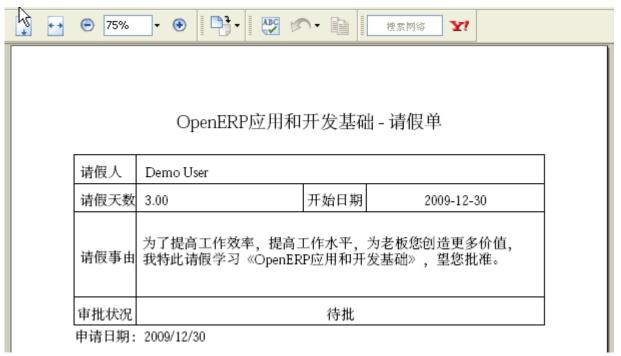


图 三-16 请假单报表

第四章 OpenERP 的权限设置

TODO

第五章 OpenERP 代码开发

TODO

第六章 超强的继承功能

TODO

第七章 基于 XML-RPC 的集成

TODO