

# 个人电话本 软件需求说明书

版本历史

2008-07-02 文档创建

## 1 引言

### 1.1 编写目的

对“个人电话本”所基于的需求作文档化

### 1.2 背景

- ◆ 软件名称：个人电话本
- ◆ 提出者、开发者：你的阳光技术频道
- ◆ 用户：使用中文的 Windows Live ID 用户

### 1.3 定义

- ◆ Windows Live ID：微软 Windows Live 登录认证服务的用户名

### 1.4 参考资料

- ◆ GB8567-88 软件文档规范
- ◆ Windows Live Dev, <http://dev.live.com>

## 2 任务概述

### 2.1 目标

“个人电话本”提供一个跨终端、私密的联系人管理网络服务。用户使用自己的 Windows Live ID 登录后，可以创建、修改、删除、查询自己的联系人；用户使用手机登录后，可以查询自己的联系人。

本产品的赢利方式是网页、移动广告收入。

### 2.2 用户的特点

本产品的定位是对计算机、浏览器、移动终端的操作略有了解的用户。每个用户的使用频率约为每周计算机登录 3 次、手机登录 10 次。

### 2.3 假定和约束

开发经费：500 元（16 工时=400 元，虚拟主机+域名=100 元）

开发期限：3 天

## 3 需求规定

### 3.1 对功能的规定

- ◆ 使用 Windows Live ID 登录，也支持 guest 体验帐号登录。
- ◆ 在网页界面创建、修改、删除联系人。每个联系人的信息包括“姓名”、“公司”、“电子邮件”、“电话”、“即时消息”、“地址”、“生日”、“我是如何认识 xxx 的？”、“自定义”、“备注”；其中“电话”、“即时消息”、“地址”的类型可以自定义成“办公电话”、“家庭电话”、“飞信”、“QQ”、“公司地址”等，而“自定义”信息的类型可以自定义成“学号”、“专业”、“身份证”等；“电话”、“即时消息”、“地址”、“自定义”信息可以输入多个，数量不作限制；对输入信息的类型和格式不作限制，长度也没有严格限定。
- ◆ 联系人通过标签作分类。每个联系人可以加一个或多个标签，例如“上海交通大学”、“人事处”、“闵行区”、“高中同学”等；可以通过标签找到带有该标签的联系人。
- ◆ 用关键字查询联系人。只要关键字出现在联系人的任意一项信息中，就能查到该联系人。
- ◆ 在网页界面呈现联系人的所有信息。手机登录时，考虑到节省流量及手机屏幕尺寸，只显示在移动情况下特别需要的“姓名”、“公司”、“电话”、“地址”几项信息。
- ◆ 在网页界面获得手机登录入口地址，并可以自动申请短地址；并设置手机登录口令。手机访问每个用户自己的入口地址，输入手机登录口令后即可用关键字查询联系人。
- ◆ 网页界面支持数据备份，将数据导出成 XML 格式。

总用户数：10000；并发用户数：100（但同一用户只能登录一次）

### 3.2 对性能的规定

精度：100%

时间特性：客户端界面响应时间 0.2 秒，服务端更新处理时间 2 秒，网络数据传送时间 5 秒

灵活性：本程序采用模块化设计，且客户端与服务端逻辑分离，可适应一定的需求变更

### 3.3 输入输出要求

### 3.4 数据管理要求

需要支持 10000 数量级的总用户数。

### 3.5 故障处理要求

### 3.6 其他专门要求

## 4 运行环境规定

### 4.1 设备

磁盘空间：初始 1MB；如要支持 10000 用户，需要 5GB 左右。

### 4.2 支持软件

操作系统：任意

Web 服务器：任意，必须配置 data 目录为 Web 不可读

脚本引擎：PHP 5，必须支持 JSON、DOM，建议支持 XSL，必须配置 allow\_url\_fopen=On

数据库服务：不需要

### 4.3 接口

Windows Live ID 登录接口，通过 yoursunny.com 转发登录实现

### 4.4 控制

软件的运行是由用户的 HTTP 请求驱动的

# 个人电话本 XML 结构设计说明书

版本历史

2008-07-02 文档创建

(由于 GB8567-88 标准中没有 XML 设计文档的规范，本文档根据 Google Code 的惯例书写)

## Audience

This document is written to state the XML file format used by PhoneBook.

## Introduction

PhoneBook uses an XML format to save user contacts data. There is no central database or SQL Server or something similar. Each user has his own XML file containing all his data.

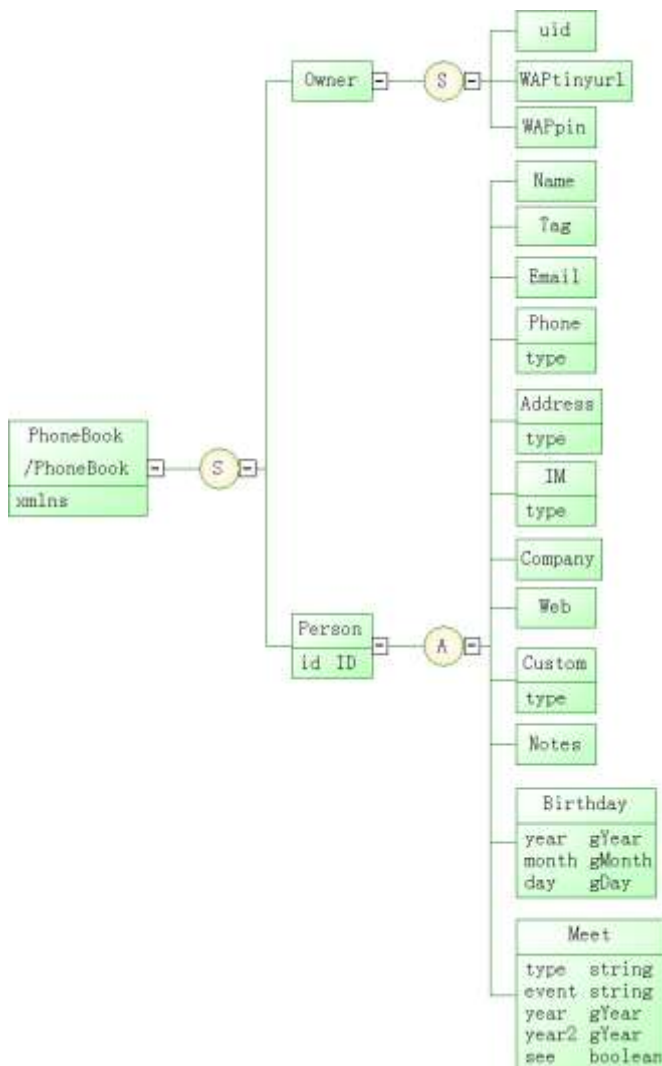
## File Name

When a user is logged in through Windows Live ID, PhoneBook application would know his ID, which is usually a 32-char string.

The file name of a user's data file is: data/id{0}/id{1}/id.xml, where id{0} means the first char of id.

## Schema

The XML Schema is defined through the following figure. You may open the XSM file with PowerDesigner 12.5 and know more.



# 个人电话本 概要设计说明书

## 1 引言

### 1.1 编写目的

对“个人电话本”的设计作文档化

### 1.2 背景

- ◆ 软件名称：个人电话本
- ◆ 提出者、开发者：你的阳光技术频道
- ◆ 用户：使用中文的 Windows Live ID 用户

### 1.3 定义

- ◆ B/S: Browser/Server, 基于浏览器和服务器的计算模式

### 1.4 参考资料

- ◆ Gmail Contacts, <http://mail.google.com>
- ◆ 海内, <http://www.hainei.com>

## 2 总体设计

### 2.1 需求规定

参见《软件需求说明书》

### 2.2 运行环境

服务器环境参见《软件需求说明书》

网页客户端支持 Firefox3、IE8 浏览器

手机客户端要求支持 WAP 上网

### 2.3 基本设计概念和处理流程

网页版本是纯 AJAX 应用。AJAX 应用直接与用户交互，接受输入、呈现数据，在必要时请求服务器。服务端脚本收到来自 AJAX 应用的请求，首先检查用户是否已登录（未登录则返回 403 错误），然后读入用户的 XML 数据文件，根据请求对 XML 数据文件进行查询或修改操作，如修改过则保存 XML 文件，以 XML 或 JSON 格式返回查询获得的数据或修改是否成功的指示。

手机版本，限于 WAP 浏览器的处理能力，所有程序处理均在服务端。WAP 浏览器发出 HTTP 请求，服务端脚本对请求中包含的用户 ID、口令、查询参数作处理，未正确登录的发送登录表单，已登录的则读取用户的 XML 数据文件，按关键字用 XPath 查询用户请求的联系人信息，用 XSLT 转换查询结果为 WML 格式，返回给手机。

### 2.4 结构

- ◆ index.htm, 网页版 HTML 网页，必要的 HTML 排版
- ◆ app.js, 网页版 AJAX 客户端脚本，包含数据模型部分、用户界面部分
- ◆ p.css, 网页版 CSS 样式表
- ◆ common.php, 服务端公用配置和公用函数
- ◆ liveid.php, Windows Live ID 登录功能服务端脚本
- ◆ pb.php, 网页版 AJAX 调用的服务端支持
- ◆ backup.php, 数据备份/导出功能服务端支持
- ◆ wap\_login.php, 手机版登录入口
- ◆ wap\_main.php, 手机版查询功能
- ◆ wap\_query.xml, 手机版查询 XSLT 转换表

### 2.5 功能需求与程序的关系

### 2.6 人工处理过程

如果服务器丢失数据，用户提交其备份的数据要求恢复，此时需要客服人员利用 Schema 人工验证数据是否符合格式，并上传到服务器。

## 2.7 尚未解决的问题

## 3 接口设计

### 3.1 用户接口

网页版使用图形界面，用 HTML、CSS、DOM 呈现。所有操作均为图形化引导。

手机版使用富文本界面，用 WML 呈现。需用户在文本框内输入，然后按功能键提交；显示提示信息后无需用户介入，自动定时跳转到下一页面。手机版的用户 ID 来自每个用户专用的入口网址、无需在文本框中输入，入口网址的形式为/wap\_login.php?u=用户 ID，并可以在网页版申请短网址以方便输入。

### 3.2 外部接口

#### 3.2.1 Windows Live ID

通过 yoursunny.com 上的转发脚本实现。

- ◆ 登录时，先请求/liveid.php；  
重定向到 <http://yoursunny.com/lib/liveid/?service=PhoneBook&serial=随机序列号&continue=liveid.php> 的完整地址；  
重定向到 Windows Live 登录页面；  
登录成功后重定向到/liveid.php?serial=随机序列号&guid=用户 ID&hash=验证字符串（验证字符串是 md5(密钥|随机序列号|用户 ID)）；  
设置登录 cookie，重定向到程序首页。
- ◆ guest 登录时，先请求/liveid.php?login=guest；  
设置登录 cookie，重定向到程序首页。
- ◆ 退出时，先请求/liveid.php?logout=1；  
删除登录 cookie；  
重定向到 <http://yoursunny.com/lib/liveid/?service=logout&continue=程序首页地址>；  
重定向到 Windows Live 退出页面；  
退出后重定向到程序首页。

#### 3.2.2 TieURL.com 短网址服务

通过文本 API 接口 GET 方式实现。

- ◆ 请求 [http://tieurl.com/api\\_text?url=手机入口的完整地址](http://tieurl.com/api_text?url=手机入口的完整地址)；  
返回申请到的短网址。

### 3.3 内部接口

#### 3.3.1 AJAX 调用的服务器接口

调用网址：/pb.php

在 HTTP 请求中，要求用 COOKIE 表示登录认证信息，这个 COOKIE 的名称是 PhoneBook\_auth、值为“用户 ID|随机序列号|sha1(密钥|用户 ID|随机序列号)”；如果没有这个 COOKIE、或登录认证信息无效，服务端应返回 403 状态码，且增加一个 header，名称是 X-PhoneBook-Login、值为 required，客户端 AJAX 应用看到这个 header 就应当提示用户登录。

对此接口的 HTTP 请求必须是 POST 方法，否则返回 400 状态码。

在 POST 字段“a”中写入请求的功能名称，支持的功能名称如下：

- ◆ a=nop。什么也不做，纯粹用于检查登录是否正确。
- ◆ a=basic。获取基本信息，用 JSON 格式返回以下数据：  
{Owner:{uid:...,WAPtinyurl:...,WAPpin:...},Tags:[{n:name,c:count},...],Phone\_types:[{n:name,c:count},...],Address\_types:[{n:name,c:count},...],IM\_types:[{n:name,c:count},...],Custom\_types:[{n:name,c:count},...]}。其中 Owner 对应 XML 文档<Owner/>标签内的三项内容，其他分别是标签、电话类型、地址类型、即时消息类型、自定义类型中已经使用的名称和各名称的运用数量。
- ◆ a=all。获取全部联系人，用 XML 格式返回。

- ◆ a=person。对一个联系人进行创建、修改、删除操作。POST 字段“id”包含被操作联系人的 id 属性（<Person/>标签的 id 属性）；如果 id 以“-”开头表示删除此 id 的联系人，如果 id 为“\_”表示分配新 id 并增加新联系人。对创建、修改联系人的操作，POST 字段“p”包含该联系人的所有属性的 JSON 表示。以 JSON 格式返回以下数据：{create:是否创建,delete:是否删除,id:被操作的联系人 id（创建时为分配到的 id）}。
- ◆ a=multi\_delete。批量删除联系人。POST 字段“id”包含逗号分隔的联系人 id 列表，这些 id 的联系人都会被删除。以 JSON 格式返回以下数据：{delete:true,id:逗号分隔的 id 列表}。
- ◆ a=multi\_tag。批量设置标签。POST 字段“P”包含联系人=>标签列表的 JSON 数据，形式为{联系人 id:逗号分隔的标签列表, 联系人 id:逗号分隔的标签列表}。以 JSON 格式返回以下数据：null。
- ◆ a=WAPtinyurl。获得短网址，如果尚未申请过则自动申请。以 JSON 格式返回以下数据：{WAPtinyurl:短网址}。
- ◆ a=WAPpin。设置手机登录口令为 POST 字段“pin”的内容。以 JSON 格式返回以下数据：null。

### 3.3.2 数据备份接口

- ◆ /backup.php，验证用户后，发送当前用户的 XML 数据文件，并令浏览器提示下载

### 3.3.3 手机访问接口

- ◆ /wap\_login.php?u=用户 ID 或 /wap\_login.php?u=用户 ID&p=错误的口令  
用户登录表单
- ◆ /wap\_login.php?u=用户 ID&p=口令  
登录成功提示消息，3 秒后自动跳转到查询表单
- ◆ /wap\_main.php?无效的登录信息  
登录无效提示消息，3 秒后自动跳转到用户登录表单
- ◆ /wap\_main.php?u=用户 ID&p=口令  
查询表单
- ◆ /wap\_main.php?u=用户 ID&p=口令&q=查询关键字  
查询结果和查询表单

## 4 运行设计

### 4.1 运行模块组合

网页版登录时使用 liveid.php，操作使用 app.js 和 pb.php，数据导出使用 backup.php。  
手机版登录时使用 wap\_login.php，查询使用 wap\_main.php。  
common.php 在服务端的所有功能中被引用。

### 4.2 运行控制

网页版客户端 AJAX 应用响应用户的鼠标、键盘输入事件，进行运行。  
网页版服务端脚本被 AJAX 应用请求时运行。  
手机版服务端脚本被 WAP 浏览器请求时运行。

### 4.3 运行时间

网页版客户端 AJAX 应用用户在用户进入页面时开始持续运行，直至用户关闭窗口。  
服务端脚本均会尽快结束并关闭文件释放资源。

## 5 系统数据结构设计

### 5.1 逻辑结构设计要点

网页版，可以直接要求服务端发回 XML（pb.php, a=all）然后解析处理，也可以用 JSON 与服务端配合工作。在客户端 JavaScript 对象中存储数据的不同形式，并根据需要填充到网页 DOM 中。  
手机版的逻辑结构只存在于服务端，且与物理结构一致。

### 5.2 物理结构设计要点

采用 XML 格式持久存储数据。参见《XML 结构设计说明书》。

### **5.3 数据结构与程序的关系**

## **6 系统出错处理设计**

### **6.1 出错信息**

网页版服务器出错时，返回 4xy 或 5xy 状态码，客户端会弹出 alert 消息框提示消息。

其他错误均没有提示。

### **6.2 补救措施**

### **6.3 系统维护设计**