



学以致用：少一点理论，多一些实践

# GNOME 3 应用程序开发

在逐阶深入的实例项目中学会GNOME应用程序开发  
领会自由开源软件（FOSS）的魅力

## 入门指南

北京GNOME用户组 协力编写



# GNOME 3 应用程序开发入门指南

北京GNOME用户组<sup>1</sup>

2014-09-11

<sup>1</sup>本书基于 CC-BY-SA 4.0 (Commons Creative Attribution-Share Alike 4.0) 许可发布。本书使用 Markdown 书写，源文件：<http://github.com/tonghuix/gnome3-app-book/>，欢迎大家提出宝贵的意见和建议，以帮助此书更加完善，方便更多中国爱好 GNOME 的朋友。



# 写在前面

本书是第一本全方位介绍 GNOME3 应用开发的书籍，自从 GNOME 基金会在 2011 年发布 GNOME3 以来，还没有如此基础和详尽介绍 GNOME3 应用程序开发的教程。

这本小册子浓缩了 GNOME 3 应用开发的主要内容，从 GNOME 的基础架构、GNOME Shell 桌面环境以及 GNOME 3 SDK 在时下流行的 Linux 发行版下的安装和搭建讲起，浅显易懂，逐阶深入，适合初学者，同时又兼顾现时流行趋向，有丰富的网络、多媒体和基于 HTML5 的应用开发例程，深入浅出，层层递进讲解，使得初学者不会像其他书那样被大量代码吓倒。

本书的一大特色是以开源软件和开源社区精神为内涵，以社区开发规范和开源软件开发规范为准绳，同时又兼顾工程化和质量体系要求；本书的形式特色是每章和每个例子后面都附有思考问题，让学习者可在引人入胜的问题中进一步深入自学，是开源爱好者和 Linux 应用开发不可多得的基础教程。

本书使用 Vala 和 JavaScript 来开发 GNOME 3 的应用程序，非常适合初级开发者。本书将会引导开发者一步步在 GNOME 3 上构建 GTK+、Clutter 和 HTML5 的应用程序。同时还覆盖了很多 GNOME 3 的独有之处，比如数据访问、多媒体支持、网络以及文件系统等，同时也很好的满足了软件工程的要求，比如本地化和测试。

关于 GNOME 即 GNU 网络对象模型环境 (The GNU Network Object Model Environment)，GNU 计划的一部分，开放源码运动的一个重要组成部分。其目标是基于自由软件，为 Unix 及类 Unix 系统构造一个功能完善、操作简单以及界面友好的桌面环境。它是 GNU 计划的正式桌面。GNOME 以及 GNOME 基金会的官方网站是：<http://www.gnome.org>——摘自维基百科。



图 1: GNOME基金会的LOGO

## 0.1 为什么写此书？

北京 GNOME 用户组作为北京地区，乃至中国大陆最大的 GNOME 中文社区，自2008年成立以来一直没有一本有关 GNOME 项目开发方面的文档、手册或书籍，虽然曾有过本地化和打包相关的活动，但这些活动对 GNOME 项目的贡献不大，且很难吸引更多成员参加。

很多新近加入的用户组成员希望可以学到一些有关 GNOME 开发相关的知识，特别是大学生。他们已经不满足于只是给 GNOME 的组件贡献本地化翻译，或者参加一些本地推

广活动，他们更希望可以深度参与到 GNOME 项目中来，希望成为国际知名开源项目中的一份子。

搜遍中国大陆出版社出版的有关 GNOME /GTK 相关的图书，大多是2000年前后出版的非常少的一两本，其版本已经非常非常老了，也许大学的图书馆角落里还能找到这样一本存书，但在市面上早就已经难觅其踪。所以没有好的教程，又如何指导开发呢？

而 GNOME 基金会在2011年发布 GNOME 3 以后，由于其用户界面、编程接口和 API 变动较大，造成大量用户和开发者流失，而本地开发者流失导致更难有人来接手相关的开发文档的编写和翻译工作。

就在此时，印尼的自由软件专家 Mohammad Anwari（默罕默德·安瓦里）<sup>1</sup> 编写的《GNOME 3 Application Development Beginner's Guide》在2012年底出版，于是我们决定着手翻译此书。

从2013年开始近一年的时间遍寻出版社出版此书的翻译版，一直未果，出版社大多以此书经济价值不高，难以收回成本等理由拒绝。对出版社来说也是无可厚非的，毕竟他们也要生存，特别是在目前竞争如此激烈的出版行业，对于这样的外版书还需要给外国出版社支付高额版权费。

因此我们决定社区自己来写，基于开源许可在互联网发布，使用《GNOME 3 Application Development Beginner's Guide》的内容，修改其原书中比较混乱的讲解，增加更多符合中文使用习惯的内容，增加本地化的翻译和一些社区相关的内容。秉承自己的文档自己写的精神，全书全部由社区成员，志愿“众包”的方式来编写，每个志愿者各负责一到两章，体现“人人为我、我为人人”的社区精神。每个人的劳动量其实并不大，但集合起来整个社区的力量则会形成难以估量的巨大价值。因此，此书也将成为中国第一本由开源社区合力编写的技术书籍。

## 0.2 各章节介绍

### • 第一章：安装 GNOME 3 及其 SDK

简单介绍了 GNOME3 的桌面环境和 SDK 的情况，并在 Fedora/OpenSUSE/Ubuntu/Debian 这些主流 Linux 发行版下进行安装和测试。

### • 第二章 基本开发环境介绍

讲述了 Anjuta IDE 的使用和设置编程环境。

### • 第三章 编程语言

通过浅显易懂的方式向读者讲述了使用命令行 JavaScript 编译器的基础知识。如数据类型、循环、数组、面向对象编程、多态、操作系统信号等。

### • 第四章 GNOME 核心库

讲述 GNOME 核心库的知识，为后面的学习进一步打下基础。

### • 第五章 构建用户界面（GUI）

---

<sup>1</sup>Mohammad Anwari（默罕默德·安瓦里），他是 GNOME 基金会成员，印度尼西亚著名的软件开发者，有 13 年的软件开发经验，在印尼生活期间创办了印尼最大的软件公司，积极参与 Linux 内核和 GNOME 相关应用软件的社区开发，有丰富的开发经验。后移居芬兰加入 Nokia 公司软件开发部，参与 QT 应用程序的开发。近几年回到印尼，专注于开发印尼的 Linux 发行版 BlackOn，并积极研发基于 GNOME Shell 桌面环境的菜单环境，最终在 2013 年 GNOME.Asia 亚洲峰会上展示，获得与会人员的一致认可。

通过 GTK+和 Clutter 组件构建基本的用户界面，脱离命令行开发，做出更有创造力的应用程序。

- **第六章 GNOME 的界面小组件 (Widgets)**

讲述如何开发界面小组件，并对其进行编程。

- **第七章 GNOME 多媒体应用程序开发**

开发 GNOME 下的多媒体应用程序，多媒体应用程序的特点和开发技巧。

- **第八章 数据对象**

这一章讨论 GNOME 数据对象 (DATA)，将会学到如何使用树形显示小组件和 EDS 服务存储数据。

- **第九章 用 GNOME 创建 HTML5 应用程序**

众所周知，HTML5 将是未来大势所趋，因此桌面环境与 HTML5 的结合势必会对桌面开发增加更多的可能，这一章是本书的特色，也是近几年 GNOME 3 所擅长之处。

- **第十章 与 GNOME 桌面集成**

这一章讲到如何与 GNOME 桌面集成起来，利用系统资源，比如 D-bus 来优化应用程序开发。

- **第十一章 应用程序的国际化与本地化**

如何让应用程序支持国际化和多语言，以利开源社区贡献者可以更加简单的翻译用户界面和本地化推广。

- **第十二章 品质控制**

这是本书的一大亮点，讲述如何开发单元测试，标准的测试流程和测试所需的工作方式。

- **第十三章 激动人心的应用案例**

本章以两个主要的大型应用程序案例 (Web 浏览器以及微博客户端) 讲述如何进行互联网应用程序开发，同时将前面学习到的知识加以综合利用。

## 0.3 需要的前期准备

阅读本书需要一些基本的面向对象编程的基础。若能有 Vala 或者 Javascript 的经验将会更加有帮助。同时，为了更好地阅读和实践书里的内容，你需要安装最新版的 Linux 发行版，比如 Fedora、OpenSUSE、Ubuntu 或其他 Debian 发行版均可。

当然，若没有面向对象编程基础也是可以阅读的，本书浅显易懂，对具有其他编程基础的人也将不会有太多难度。

## 0.4 本书的目标读者

本书适合那些希望以 GNOME 3 作为目标平台的软件开发者。同时还兼顾那些希望创建跨平台应用程序的开发者，因为毕竟 GNOME 3 中的很多库是横跨 Linux、OS X 和 Windows 多平台系统的。

## 0.5 本书的一些约定

本书当中会有一些格式约定。一些常见的约定，列举如下：

---

### 0.5.1 实践环节

1. 第一步
2. 第二步
3. 第三步

每一步一般也需要一些解释，所以常常会跟着这种：

#### 刚才发生了什么

这将会解释刚才究竟发生了什么情况。当然你也会看到本书其他的一些辅助信息：

#### 小测试

这将会是一个多选题，将会帮助你更好地理解一些学到的知识。

#### 大胆实践

非常大胆的实践机会用来帮助你来实践刚刚所学的知识。你会发现不同的信息，其字体也会有相应的变化。比如：

文字中的代码会印刷成类似这样：“可通过修改 `configure.ac` 文件来包含 `WebKitGTK` + 库到此项目中。”

代码块将会标示成下面这样：

```
1 using Glib;
2 using Gtk;
3 using WebKit;
4 public class Main : WebView
5 {
6     public Main ()
7     {
8         load_html_string("<h1>Hello</h1>","/");
9         // 代码注释
10    }
11    /* 更多注释
12     * some comments
13     */
14    static int main (string[] args)
15    {
16        Gtk.init (ref args);
17        var webView = new Main ();
18        var window = new Gtk.Window();
19        window.add(webView);
20        window.show_all ();
```



```
21     Gtk.main ();
22     return 0;
23 }
24 }
```

任何命令行的输入输出将会表示成：

```
LANGUAGE=id LC_ALL=id_ID.utf8 src/hello-i18n
```

**新的概念**和**重要的词语**将会加粗提示。任何屏幕提示或者菜单选项或者对话框等，将会显示成这样：“点击**继续**，完成对 **GTranslator** 的设置并通过其菜单打开 **id.po** 文件。”

---

警告或者重要提示将以缩进的方式显示。

---

一些小窍门也将如此显示

## 0.6 致谢

首先感谢 Mohammad Anwari（默罕默德·安瓦里）于2012年编写了《GNOME 3 Application Development Beginner's Guide》一书，将 GNOME 3 应用程序开发的要义写明出来。我们在原书基础上大量增加新的代码实例和大量的本地化内容，丰富和加强了原书的架构，更适合中国程序员阅读，同时降低了入门难度。

同时还要特别鸣谢 LarrycaiYu，本书的**源代码**<sup>1</sup>就是基于其 **sdcamp**<sup>2</sup> 项目发展而来。LarrycaiYu 利用 MultiMarkdown 和 Latex（xelatex）将书写中文技术书籍变得非常简单，同时将其与Git结合，并结合GitHub的社交编程功能，使得多人协作编写技术书籍更加易如反掌！

本书的编写得到了北京GNOME用户组大量志愿者的热心参与，其中：

- **佟辉** 完成本书源文件架构的初始化，并编译了第一章“安装 GNOME 3 及其 SDK”。

希望各位朋友在阅读本书的同时，能够积极投身到自由软件和开源软件的大潮中。随时关注北京 GNOME 用户组的网站，参加北京 GNOME 用户组的活动，进而贡献到 GNOME 全球社区中来。



北京 GNOME 用户组 网站: <http://www.bjgug.org>

北京 GNOME 用户组 微博: <http://weibo.com/bjgug>

---

<sup>1</sup><https://github.com/tonghuix/gnome3-app-book/>

<sup>2</sup><https://github.com/larrycai/sdcamp>



# 目录

写在前面	i
0.1 为什么写此书?	i
0.2 各章节介绍	ii
0.3 需要的前期准备	iii
0.4 本书的目标读者	iii
0.5 本书的一些约定	iv
0.5.1 实践环节	iv
0.6 致谢	v
目录	vii
1 安装 GNOME 3 及其 SDK	1
1.1 系统需求	1
1.1.1 GNOME Shell	2
1.1.2 GNOME Panel /GNOME Classic Mode	2
1.1.3 开发需求	2
1.2 GNOME 3 桌面的主要架构	2
1.3 GNOME 及其 SDK	5
1.3.1 实践环节 —— 在 Debian Testing 上安装 GNOME 及其 SDK	5
1.3.2 实践环节 —— 在 Ubuntu 14.04 上安装 GNOME 及其 SDK	7
1.3.3 实践环节 —— 在 openSUSE 13 下安装 GNOME 及其 SDK	8
1.3.4 实践环节 —— 在 Fedora 20 中安装 GNOME 及其 SDK	9
1.4 小结	11
附录 A 习题解答	13



---

## 第 1 章

# 安装 GNOME 3 及其 SDK

GNOME 3 是自 GNOME 项目 1999 年发布以来，最新也是最强大的版本。GNOME 3 在之前版本的基础上为桌面环境的用户体验带来了突破性变化，搭载了更加流畅的用户界面和动画特效，它完美诠释了现代计算机通过使用 GNOME Shell 将会为新的用户体验（UX）带来如何飞跃性的改进。当然，对那些习惯了传统用户体验来说，依旧保持了 Classic Mode (传统模式) 以兼容硬件配置不支持的情况。

GNOME 因其简约质朴的风格被大家所熟知，目前已被大量的主流 Linux 发行版选为它们的默认的桌面环境。GNOME 也是**第一个拥有大量用户界面指南的自由软件**，同时还能保持安全性、可用性以及用户友好。

GNOME 带来的巨大变革，使信息产业也因此有了深远的变化，使其成为了世界上最先进也是最具实用性的自由桌面。与此同时，它通过引进大量的先进技术到系统中，最大化利用硬件的同时让用户在使用计算机时享受最美好的体验。

在我们正式开始前，还是需要安装一下 GNOME 桌面以及其开发环境到我们的电脑中。当然作为初学者，我们将会演示在一些主流的 Linux 发行版下的操作。特别的，本章我们将会涉及到这样一些话题：

- 系统要求
- 本书当中涉及到的 GNOME 3 基本体系结构
- 安装 GNOME 3 及其 SDK 到各种主流的 Linux 发行版中。

好了，让我们开始吧！

### 1.1 系统需求

GNOME 3 提供了两种不同的**用户体验**（UX，User Experience）以应对两种不同的硬件环境—— GNOME Shell 以及 GNOME Panel /Classic Mode。这两种不同的 UX 有不同的需求，不过最基本的需要：

- 800MHz CPU（至少 1GHz 以达到比较好的性能）
- 512MB 内存（至少 1GB 内存可以达到最佳性能）
- 至少 2GB 的硬盘空间（如果你想安装更多应用就需要更大的空间）

### 1.1.1 GNOME Shell

这是 GNOME 3 中最新增加的组件，提供了炫酷的用户体验。此 UX 需要你安装的显卡支持 OpenGL 可用的 3D 加速。当然目前大多数显卡厂商比如 Intel、ATI 和 Nvidia 都可以很好的运行<sup>1</sup>。

然而，即便你已经很清楚的知道显卡可以支持 OpenGL，还需要确认你已经安装并启用了合适的显卡驱动程序，否则系统也不会允许你使用这个 UX。

---

你可以通过[h-node](http://www.h-node.org/videocards/catalogue/en)页面检查显卡是否兼容 GNOME Shell: <http://www.h-node.org/videocards/catalogue/en>，同时检查是否有3D加速。

---

### 1.1.2 GNOME Panel /GNOME Classic Mode

这是传统的GNOME桌面形式，提供了比较基础和较少的吸引人的用户体验。当系统无法启动 OpenGL 的时候会返回到这个模式（Fallback）。这个模式与老版本的 GNOME 非常相似，但是其用户界面已经修改了很多以配合 GNOME Shell 的用户界面，比如主 UI 就已经完全不同，但其锁屏界面与之前还是挺相似的。

当然幸运的是，本书当中绝大部分内容都不涉及太多的 GNOME Shell 的深度应用，因此我们完全可以非常安全的运行在 GNOME Panel /GNOME Classic Mode 下面，对学习本书的内容并不会产生什么影响。

### 1.1.3 开发需求

作为我们进行软件开发来说，需要更多的计算机资源、内存啊、CPU啊，以及更多的存储空间。以下是一些比较好的系统需求，可以让你的开发过程轻松一些：

- 多核 2GHz CPU
- 4GB 内存
- 500GB的硬盘空间，最好是SSD固态硬盘

好消息是我们可以很简单的在虚拟机上开发 GNOME 3 应用程序，在虚拟机上安装不同的 Linux 发行版，我们可以切换各个虚拟机并跟踪各种问题，以使我们开发的 GNOME 3 应用程序可以在各个发行版都能流畅运行。使用 GNOME Shell 的话，你需要确保你已经在虚拟机程序中启用了 3D 加速选项。

比如在 VirtualBox 中，我们可以打开虚拟机的 设置 对话框，点击 显示 选项卡，并勾选 启用 3D 加速。记住，如果你的宿主机没有相应的 3D 加速功能，这个选项是不能启用的。

## 1.2 GNOME 3 桌面的主要架构

当我们谈到 GNOME 3 的时候，很多人只认为是 GNOME Shell，这是不对的。GNOME Shell 只是整个 GNOME 桌面架构中的一个组成部分，甚至我们可以替换掉它（比如可以用 GNOME Panel /GNOME Classic Mode）或者干脆移除。

---

<sup>1</sup>目前的一个可能的难点是显卡驱动不支持，不过大多数情况都比较容易解决，如果碰上无法运行的情况可以反馈到 GNOME 官方的邮件列表中。——编者注



图 1.1: VitruaBox 中对显卡 3D 加速的设定

事实上，GNOME 的概念远大于 GNOME Shell。它提供了应用程序的一套基础架构，这样应用程序就可以和系统通信、渲染文字、播放动画效果、获取数据等等。在开始开发以前，我们最好理解一下 GNOME 的主要架构，同样也可以帮助我们知道哪些需要安装。让我们先从以下这个 GNOME 架构框图开始：

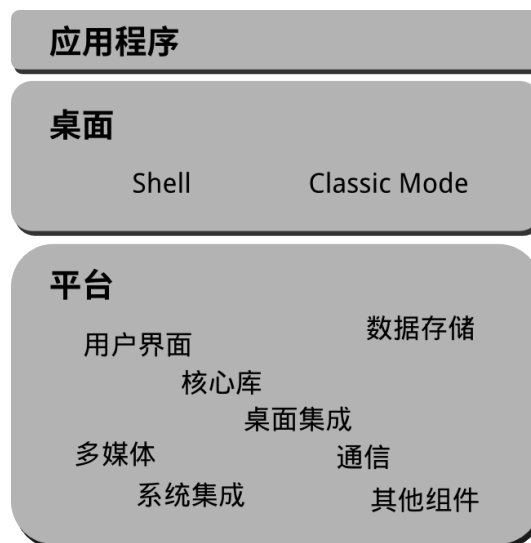


图 1.2: GNOME 架构框图

从此架构框图我们就可以看出，GNOME Shell是应用程序的基础，同时又在 GNOME 平台架构之上。本书将会涉及主要的平台架构并在一定程度上触及一些上层内容（应用程序层和桌面层）。

有关 GNOME 平台更多深层相关的内容可以参考 GNOME 开发者网站，<http://developer.gnome.org>。

下面就具体来介绍一下 GNOME 平台的各个组件，特别是本书中将会涉及到的内容：

- **核心库：**这是 GNOME 架构中最底层的接口，主要包括了：
  - **GObject：**这是 GNOME 的对象系统。主要是 GNOME 中用 C 语言实现了一些面向对象的方法。
  - **GLib：**此通用库主要包括了 GNOME 架构都会用到的一些内容。
  - **GIO：**这是一个虚拟文件系统库，提供了文件访问、存储卷以及抽象化的驱动程序。
- **用户界面库：**此工具集用来构建图形应用程序，并且包括了：
  - **GTK+：**也就是历史上著名的 **GIMP Toolkit**<sup>1</sup>，这是 GNOME 中默认用来构建图形应用程序的工具集。它提供了一系列小部件和工具。
  - **Cairo：**此库帮助绘制 Canava 组件。主要用它来创建新的小部件（Widgets）或者对已经存在的小部件进行扩展开发。
  - **Pango：**此库主要用来进行文本渲染。
  - **ATK：**这是用来开发无障碍访问（Accessibility，A11y）的工具。它为一些特殊用户在使用 GNOME 的时候提供一些不同的帮助。比如针对有视力障碍者提供屏幕放大和文字朗读等。
  - **Clutter：**此工具用来创建具有大量且平滑动画特效的应用程序。需要 OpenGL 支持。
  - **Webkit：**这是 Web 工具集。它提供了全功能的并可兼容 HTML5 文档的渲染引擎。
- **多媒体库：**提供了播放和校验多媒体文件的功能，包括了：
  - **GStreamer：**一个强大的多媒体库
- **数据存储：**提供了一些数据访问相关的库：
  - **Evolution Data Storage (EDS)：**主要包括了 libebook 以及 libecal，并提供了接口，用以访问 Evolution 管理的通讯簿以及日历。

本书中我们也需要用一些工具，主要有：

- **Seed：**这是 GNOME 世界中的 JavaScript 集成。我们将会用 Seed 来开发应用程序脚本。当我们安装 GNOME Shell 的时候会自动安装好 Seed，因此一会我们不会特意介绍如何安装。
- **vala：**这是一个新兴的面向对象编程语言，主要用于开发 GNOME 应用程序。
- **Anjuta：**这是 GNOME 下的一个集成开发环境（IDE）。
- **Glade：**这是一个 GTK+ 的图形界面设计工具。
- **Gtranslator：**这是一个用来翻译应用程序用户界面到本地语言的工具。

---

<sup>1</sup>最初是GIMP的专用开发库（GIMP Toolkit），后来发展为类Unix系统下开发图形界面的应用程序的主流开发工具之一。GTK+是自由软件，并且是GNU计划的一部分。GTK+的许可协议是LGPL。GTK+使用C语言开发，但是其设计者使用面向对象技术。首个稳定版于1998年4月14日推出。——摘自维基百科。



- Devhelp: 这是一个文档引用查询工具，真的可以很方便的帮助我们开发。

---

注意。这些工具的包名并不是标准的，因此你会发现实际安装的时候会与以上略有不同。不用担心，相关章节都会有涉及。

---

## 1.3 GNOME 及其 SDK

我们将会安装 GNOME 及其软件开发套件 (SDK, Software Development Kit) 到你最喜欢的 Linux 发行版上，会涉及一些主流发行版。我们不会讲每种发行版的安装过程，因此希望你已经拥有并在使用一些 Linux 发行版，熟悉其基本操作。我们只是安装一些额外的包以获得 GNOME 及其 SDK。

你可以直接跳过其他发行版的安装过程，只关注你喜欢的即可。如果你用的发行版并没有列出来，也不用担心，只需要找找相近的发行版，并用搜索工具搜索一下包名试试。现在我们就要开始安装了，该是动手实践的时刻了。

---

### 1.3.1 实践环节 —— 在 Debian Testing 上安装 GNOME 及其 SDK

目前的 Debian Testing 版本是 Jessie (8.0)，默认使用 GNOME 3 桌面环境，因此我们只需要安装 SDK 即可。Debian Testing 下我们可以打开 **软件包** 工具来安装。

1. 点击屏幕左上角的**活动**按钮。
2. 点击**显示应用程序**图标
3. 点击**软件包**图标（或者在搜索栏里输入“软件包”）。
4. 之后就可以看到这样的画面：



图 1.3: Debian Testing 下的软件包工具

## 刚刚发生了什么？

软件工具是一个先标记后安装的软件包管理器，也就是说在我们真正安装前，我们有机会去改变选择安装的包。在我们点击**应用（Apply）**以后才会开始安装。也正是因为有这样的特性，我们可以让一边安装一边浏览和搜索，不用等待冗长的安装过程结束，再去安装下一个包。

如图1.3所示，这个工具有两栏，左侧是搜索和选择包的类别，右侧则会显示搜索结果（或类别里的内容），下边的方框内会列出选定软件包的描述。

## 标记 SDK 软件包

SDK 软件包一般是被包含在各种开发包中。所谓开发包则包含了相应开发所需的头文件，用来编译和链接。如果平时仅仅只是运行与库编译链接好的程序，则不需要这些包。

这些开发包往往会冠以 **lib** 或以 **-dev** 结尾，他们通过内部的依赖关系与相应的库绑定在一起，因此无论何时安装这些开发包，包含相应库的软件包也会一并自动安装。

表1.1列出了 GNOME 组件在 Debian Testing 上的包名称，我们只需要搜索并标记上即可。

表 1.1: Debian Testing 下 GNOME 3 SDK 主要软件包列表

子系统	包名称
核心库	libglib2.0-dev (GIO 和 GObject 已经包括在此包里)
	libglib2.0-doc
用户界面	libgtk-3-dev
	libgtk-3-doc
	libcairo2-dev
	libcairo2-doc
	libpango1.0-dev
	libpango1.0-doc
	libatk1.0-dev
	libatk1.0-doc
	libclutter-1.0-dev
	libclutter-1.0-doc
	libwebkitgtk-3.0-dev
	libwebkitgtk-3.0-doc
多媒体库	libgstreamer0.10-dev
数据存储	libecal1.2-dev
工具和基础开发包	valac-0.16
	anjuta
	glade
	gtranslator
	devhelp

## 应用并安装

在标记完之后，我们就可以点**应用**按钮来安装了。稍事休息一下，等待安装完成，之后我们就可以进行下一章的内容了。

另外，熟悉 Debian 的朋友亦可以使用命令行下的 `aptitude` 工具或者 `apt-get` 等工具来安装。此处不赘述。

### 1.3.2 实践环节 —— 在 Ubuntu 14.04 上安装 GNOME 及其 SDK

Ubuntu 默认使用了自己的桌面环境，也就是 **Unity**。我们安装 GNOME 3 桌面环境及其 SDK 利用 Ubuntu 软件中心：

1. 在屏幕左侧的边栏中点击 Ubuntu 软件中心。
2. 点中搜索框，键入 `gnome-shell`
3. 在搜索的第一个结果旁点击**安装**按钮
4. 稍等片刻，GNOME Shell 就装上了。

刚刚发生了什么？

刚刚我们只是安装了 GNOME Shell 桌面，因为 GNOME Panel /Classic Mode 与 GNOME Shell 存在依赖关系，因此现在把两种 UX 都安装好了。

Ubuntu 软件中心并不是先标记后安装的，因此我们需要每次搜索一个包安装一个包，而安装进程则是在后台进行的，用户不能直观看到，更糟糕的是后我们无法一边安装一边搜索，不过呢若搜索出来的结果恰好在一页里，我们可以点击安装按钮让其排队自动完成。

若想在 Ubuntu 下开发 GNOME 应用程序，这里推荐使用 **Ubuntu GNOME 发行版**<sup>1</sup>。此发行版已经默认集合了 GNOME 桌面，同时更具备 Ubuntu 的一些优势。因此只需要直接安装 GNOME 3 的 SDK 就行了。

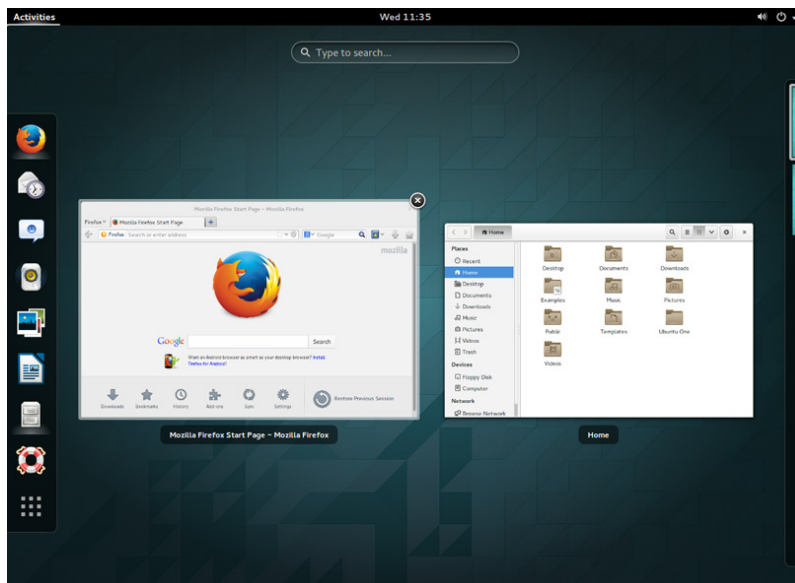


图 1.4: Ubuntu GNOME 14.04 LTS

<sup>1</sup><http://ubuntugnome.org/>

## 继续安装 SDK

Ubuntu 是基于 Debian 的，因此我们可以使用如表1.1相同的包名称来安装，现在只需要一个包一个包的来安装就行了。安装完成之后，我们就可以进入到下一章的学习中啦！

与 Debian 一样，在 Ubuntu 下也可以使用命令行下的 `aptitude` 工具或者 `apt-get` 等工具来安装。此处不赘述。

### 1.3.3 实践环节 —— 在 openSUSE 13 下安装 GNOME 及其 SDK

openSUSE 13 已经默认安装 GNOME（若安装的是 KDE 桌面或其他桌面，也可以通过下面的 YaST 工具来安装），因此我们只需要关心如何安装 SDK 即可。为了管理各种软件，openSUSE 提供了 YaST (Yet another Setup Tool) 工具，按下面的步骤就可以在 openSUSE 13 下安装 GNOME 及其 SDK。

1. 点击屏幕左上角的活动按钮
2. 点击显示应用程序图标
3. 再打开 YaST 工具即可（或者在搜索栏键入 `yast`），之后会要求你输入正确的 root 密码以验证身份
4. 这样 YaST 就打开了，点击软件管理图标。
5. 等待其更新软件包和数据库之后，就可以看到如图1.5的界面了：



图 1.5: openSUSE 13 提供的软件管理工具

## 刚刚发生了什么？

YaST 是 openSUSE 提供的一系列系统管理工具的集合，软件管理是其中之一。这个工具有两栏，左侧是软件包的类别，右侧是选择的类别下的软件包列表或者搜索的结果，以及软件包的描述。

与 Debian 下的软件包工具类似，YaST 也是先标记后安装的，这就意味着我们需要先勾选需要安装的包，然后再安装。确保所有我们要安装的包都勾选上以后，最后就可以点击安装了。

勾选 SDK 软件包

SDK 软件包一般是被包含在各种开发包中。openSUSE 使用 RPM 包管理系统，开发包的后缀名称是 `-devel`，同时相应的库的软件包会随开发包包的同时一起安装。

表1.2列出了 openSUSE 13 下的 GNOME 组件的包名称。我们可以一一搜索这些包，并将之勾选上。

表 1.2: OpenSUSE 13.1 下 GNOME 3 SDK 主要软件包列表

子系统	包名称
核心库	glib2-devel (GIO 和 GObject 已经包括在此包里)
用户界面	gtk3-devel gtk3-devel-docs libseed-gtk3-devel cairo-devel pango-devel atk-devel clutter-devel clutter-doc libwebkitgtk3-devel
多媒体库	gstreamer-0_10-devel gstreamer-0_10-docs gstreamer-0_10-fluendo-mp3
数据存储	evolution-data-server-devel evolution-data-server-doc
工具和基础开发包	vala anjuta glade (不要与 glade3 弄混) gtranslator devhelp

开始安装

全部都勾选完毕以后，就可以点**应用**按钮来启动安装了。等待一段时间安装完成，我们就可以进入到下一章的学习了。

1.3.4 实践环节 —— 在 Fedora 20 中安装 GNOME 及其 SDK

Fedora 20默认搭载 GNOME，因此我们主要集中在安装 SDK。要想在 Fedora 上安装软件包，可以运行**软件**工具。但其不能帮我们安装 SDK，因此只能通过命令行的工具来完成了。

- 1. 点击屏幕左上角的**活动**按钮
- 2. 点击**显示应用程序**图标
- 3. 点击**终端**图标（或者在搜索栏键入terminal）

4. 打开终端输入命令。
5. `yum search <包名称>` 此命令可以用来搜索软件包，以确定是否安装
6. `sudo yum install <包名称>` 此命令用来安装软件包，需要 root 权限。

### 刚刚发生了什么？

SDK 一般是放在所谓的开发包中的。Fedora 与 openSUSE 一样都是 RPM 包管理系统。开发包的包名是以 `-devel` 结尾的，而实际上的库则是在包名没有 `-devel` 的软件包里面的，当安装 `devel` 软件包的时候，也会把同名的库一并安装。比如 GLib 在 `glib2` 包内，其开发包则是 `glib2-devel`，无论什么时候安装 `glib2-devel`，`glib2` 都会自动安装。

表1.3列出了 Fedora 下的 GNOME SDK 软件包名称。

表 1.3: Fedora 20下 GNOME 3 SDK 主要软件包列表

子系统	包名称
核心库	<code>glib2-devel</code> (GIO 和 GObject 已经包括在此包里)
用户界面	<code>gtk3-devel</code> <code>gtk3-devel-docs</code> <code>cairo-devel</code> <code>pango-devel</code> <code>atk-devel</code> <code>clutter-devel</code> <code>clutter-doc</code> <code>webkitgtk3-devel</code> <code>webkitgtk3-doc</code>
多媒体库	<code>gstreamer-devel</code> <code>gstreamer-devel-docs</code>
数据存储	<code>evolution-data-server-devel</code> <code>evolution-data-server-doc</code>
工具和基础开发包	<code>vala</code> <code>vala-doc</code> <code>vala-tools</code> <code>anjuta</code> <code>glade</code> (不要与 <code>glade3</code> 弄混) <code>glade-libs</code> <code>gtranslator</code> <code>devhelp</code>

### 完成安装

如上所说，我们可以利用 `yum search <包名称>` 命令先搜索出来这些包是否安装。然后再用 `sudo yum install <包名称>` 来安装。其实也可以在一条命令里同时安装多个包，比如：

```
sudo yum install glib2-devel gtk3-devel gtk3-devel-docs cairo-devel
```

所有相关包都安装完毕以后，就可以跳到下一章了。

## 1.4 小结

本章我们学习了一些有关 GNOME 的基础理论知识，了解了其系统需求，基础架构和 GNOME 及其 SDK 的安装。特别的，我们知道了 GNOME 3 提供了两种不同的 UX，我们也懂得了其系统需求是不同的，以及开发所需的系统需求。还有一些小窍门，比如我们可以在虚拟机环境中运行。还讨论了 GNOME 的基础架构以及本书中将会涉及到的每一个组件的功能和作用。

我们现在也知道了 GNOME 的组件在架构图框中的位置。当然同时现在也知道如何在主流的几大 Linux 发行版上安装 GNOME 的软件包和开发包，这让我们领略了不同发行版之间的差异。

千里之行始于足下，这章的学习为后面学习 GNOME 应用程序开发做好了铺垫。下一步就要准备我们的开发工具了，我们将要学习如何使用它们，以及接触一点点 Vala，也就是本书将会涉及到的一种编程语言。





---

## 附录 A

## 习题解答

TBD