

## 如何托管CydiaTM存储库

几个月前，[我介绍了Cydia和Telesphoreo](#)。从那时起，我收到了一些关于如何创建Cydia“源”的问题。许多人认为Cydia使用与AppTapp Installer相同的存储库结构，但当时情况绝对不是这样，因为它不会解决AppTapp的[打包问题](#)。幸运的是，这个问题的答案通常很短：Cydia存储库是Debian APT存储库。

然而，尽管Debian APT有[很好的记录](#)，但这对一些人来说还不够，部分原因是官方文档通常会转移到存储库的所有转移。此外，Cydia确实对基础结构有一些值得记录的扩展。因此，我整理了这份文件，以帮助引导人们进入使用Cydia使用尽可能简单的基础设施分发软件的世界。

### 第1步：制作包裹

在Debian APT/dpkg的世界里，你可以安装的任何东西都是“软件包”。这些软件包以.deb文件的形式存在，这些文件具有相当神秘的内部格式（在软件历史学家中，这可以被理解为“迷入”），本指南不会让读者感到厌烦。在高级级别上，.deb包含将作为软件包的一部分安装的文件，以及包含软件包元数据（其名称、大小和其他杂项详细信息）的“控制”信息。

此类文件的构建使用工具dpkg-deb完成。Debian（或Macintosh上的Fink，也基于APT）的用户已经安装了此软件包，但几乎所有其他平台的用户都可以轻松安装它，也可以在他们的iPhone上完成这项工作。为了为少数顽固的人加快这个过程，我为[Linux](#)和[Mac OS X 10.4+](#)编译了静态链接副本。

然后，要制作一个软件包，我们只需要准备一个文件夹，其中包含我们要安装的文件，因为它们将显示在iPhone的文件系统上。此外，我们将在软件包的根目录中添加一个名为DEBIAN的目录，以存储一个名为“控件”的文件，该文件将包含我们的元数据。例如，如果我们想为MyProgram安装程序和LaunchDaemon，我们（当然取决于我们安装的内容）最终可能会采用以下目录结构：

```
+-- MyProgram+--应用程序+-- MyProgram.app| +-- Info.plist+-- MyProgram
```

控制文件的内容是一系列名称/值对（用冒号分隔），每行一个。以下是这样一个完整的控制文件的示例，以及对每个字段的含义及其包含的值类型的分解描述。

```
<saurik@saurik.com>赞助商：微软<http://www.microsoft.com/>部分：游戏
```

- **软件包：**这是软件包的“标识符”。在小写中，这应该是一个反向主机名（就像苹果Info.plist文件中的“捆绑标识符”）。如果您还选择托管AppTapp安装程序存储库以支持传统客户端，[强烈](#)建议您使此名称与AppTapp捆绑标识符匹配（小写除外）。
- **名称：**当软件包显示在Cydia的列表中时，有一个更漂亮的名称很方便。此字段允许您用任意字符串覆盖此显示。此字段可能会经常更改，而“软件包”字段在软件包的生命周期内是固定的。
- **版本：**软件包的版本表示两个独立的值：软件包中的软件版本和软件包本身的版本。这些版本号由连字符分隔。
- **架构：**这描述了软件包是为哪个系统设计的，因为从iPhone到台式计算机的所有内容都使用.deb文件。iPhoneOS 1.0.x/1.1.x的正确值是“darwin-arm”。如果您要部署到iPhoneOS 1.2/2.x，则应使用“iphoneos-arm”。
- **描述：**这个栏位比其他栏位复杂一点，因为它可能使用多行。第一行（冒号之后）应该包含一个简短的描述，显示在软件包名称下方的软件包列表中。或者，可以选择将该描述替换为任意长的描述，该描述将显示在软件包详细信息屏幕上。从技术上讲，此字段的格式相当复杂，但Cydia目前忽略了大部分复杂性：相反，Cydia允许您在此字段中放置任意令人敬畏的HTML。此扩展描述的每一行必须以空格开头。然而，我强烈建议将此用于HTML：您应该使用Depiction：用于Cydia中的描述，并使用扩展描述（然后Cydia将忽略），以与命令行客户端兼容。我通常会让这个栏位子没有记录在案，但这与APT/dpkg太不同了，我觉得有必要在这里做一个完整的解释。可以说，在未来的某个时候，我应该在Cydia中重做这个字段以正确解析，所以另一种看待这个问题的方式是，扩展描述根本不应该与Cydia一起使用。
- **主页：**通常，包装商想要提供的关于软件包的信息比软件包描述中列出的信息还要多。Cydia支持详细信息屏幕上的“更多信息”字段，该字段将用户分流到打包程序选择的网站。
- **描述：**Cydia的整个界面几乎是一个网页，这使得远程添加功能或新功能变得非常容易。您可能想要的一件事是能够显示具有特殊格式的自定义链接或屏幕截图.....只需在您的软件包页面上显示一些特别的东西（甚至是广告）。这是用“描述”完成的，这是一个加载到iframe中的URL，取代了通常预设的描述：和主页：链接。有关一个很好的例子，请参阅Cydia中的WinterBoard软件包详细信息页面。对于许多打包商来说，这已成为他们的更多信息页面，仅用于向后兼容。然而，它不需要是一样的。如果您包含Depiction:，您也可以考虑根本没有主页：字段。
- **维护者：**构建软件包的人被称为“维护者”。这是将就包裹本身相关问题与之联系的人。这应该是“他们的姓名<email@adddre.ss>”的形式。
- **作者：**相比之下，编写原始软件的人被称为“作者”。此名称将显示在详细信息屏幕上的包裹名称下方。该字段的格式与“维护者”相同。
- **赞助商：**最后，可能有人只是提供影响力或现金来实现包裹。此人应以“维护者”的形式在这里列出，除非使用资源URI而不是电子邮件地址。
- **部分：**在Cydia的“安装”选项卡下，软件包按“部分”列出。如果您想将空格编码到您的章节名称中，请使用下划线（Cydia将自动转换这些下划线）。

然后，通过前往包含MyProgram的文件夹并运行dpkg-deb来构建软件包。结果是一个可以在iPhone上安装和测试的.deb文件。根据您使用的dpkg-deb版本，您可能会收到一些关于“用户定义字段”的警告。这是由于Cydia在控制字段格式中添加的扩展，假设它们看起来正确，可能会被安全地忽略。

如果您使用的是Macintosh，在执行这些步骤之前，您需要再做一件事。当在苹果系统上创建tar文件（Debian软件包内部结构的一部分）时，会创建一些包含资源fork信息的额外\_\*文件。然后，这些文件夹与软件包一起安装，可能会与其他软件包冲突（真的根本不应该存在）。要关闭它，您需要导出以下环境变量：

```
导出 COPYFILE_DISABLEexport COPY_EXTENDED_ATTRIBUTES_DISABLE
```

```
[root@desktop:~/cydia]# dpkg-deb -b MyProgram警告，'MyProgram/DEBI
```

显然，这只是关于您可以使用APT做什么的冰山一角。在安装过程中，您的软件包可以使用shell脚本连接许多不同的点，您可以将文件标记为“配置文件”（只有在有意义时才进行智能管理和升级），等等。有关更多信息，请扫描到本文底部以获取外部资源（本文不讨论这些功能，因为95%的软件包不需要此类内容）。

### 第2步：测试软件包

现在我们已经构建了软件包，谨慎的做法是将其复制到手机上进行测试。由于.deb文件完全是自包含的，因此这是一个相对简单的过程。在手机上，可以使用“dpkg -i”直接从文件中安装它。安装后，它可能由Cydia管理（Cydia将显示软件包来自源“本地/未知”）或使用apt-get进行操作。

```
iPhone:~ root# dpkg -i MyProgram.deb选择之前取消选择的软件包com.sauri
```

### 第3步：创建存储库

该过程的最后一步是创建一个存储库，其他用户不仅可以安装，还可以通过Cydia找到您的软件。此步骤需要在某个网站的某个地方创建一个文件夹，其中包含两个文件：您的新.deb和一个名为Packages的索引。此索引使用名为dpkg-scanpackages的Perl脚本创建，该脚本使用目录的名称来扫描.deb文件。在这里，我们传递-m表示“所有文件，包括重复文件”和/dev/null作为“覆盖”文件（这将导致一些我们可能会忽略的警告）。

```
[root@desktop:/web/apt/xmpl]# dpkg-scanpackages -m ./dev/null >软
```

不幸的是，Debian附带的dpkg扫描包的库存副本没有考虑到用户定义的字段，因此需要手动将Cydia特定的字段添加到脚本中。这相当简单：只需在数组“fieldpri”的末尾添加“名称”、“作者”、“主页”和“图标”。如果您愿意，您可以[从我的网站](#)下载此文件的预制版本。

不幸的是，我提供的这份副本不适用于芬克，所以你必须自己做相当简单的修改。如果我很快拿到股票芬克副本，我可能会在这个网站上提供修改过的副本，但目前我身上没有它:({

目前，此程序可能无法在iPhone上运行，因为Perl尚未移植到Telesphoreo。希望作者在不久的将来解决这个问题，届时dpkg的一些更高级的功能将最终解锁（例如dpkg-divert，它允许从其他软件包或系统中安全替换文件）。

### 第4步：存储库元数据（可选）

存储库的另一个功能是有关运行它的人的信息，它的名字等。此信息存储在名为Release的文件中，该文件格式如下。此文件是严格可选的。如果您将其关闭，那么Cydia会从您的URL中合成信息，以显示源上的详细信息。

```
起源：Saurik的Cydia示例标签：Cydia示例套房：稳定版本：0.9代号：tangelo架
```

- **来源：**Cydia将此用作存储库的名称，如源编辑器（和其他地方）所示。这应该是对存储库的更长但不要太长的描述。
- **标签：**在软件包列表屏幕上，Cydia显示存储库和部分软件包来自哪个。然而，此位置没有太多的空间，因此此字段应该包含存储库名称的较短/更简单版本，可以用作标签。
- **套件：**只需将此设置为“稳定”。这个字段可能不是必需的，但谁知道呢？我，肯定不会。
- **版本：**这是一个任意版本号，实际上没有解析。我要看看它有多需要。
- **代号：**在“自动”存储库中，您可以为不同的目标系统存储多个软件分发。例如：apt.saurik.com的主要存储库包含桌面Debian Etch系统和iPhone的内容。然后，这个代号描述了我们目前正在寻找的发行版。在“琐碎”的存储库中（如本文所述），您可以将您想要的任何内容放在这里，该字段甚至可能是可选的。
- **架构：**要验证存储库是否适用于您正在使用APT的特定设备，请在此列表的发布文件中查找。您必须在这里指定软件包文件中显示的所有架构。同样，我们使用darwin-arm用于1.1.x，iphoneos-arm用于2.x。
- **组件：**只需将此设置为“主”。这个字段可能不是必需的，但谁知道呢？我，肯定不会。
- **描述：**软件包源屏幕上列出了存储库的简短描述。此描述最终可能与软件包相似（具有长/短品种和上述编码），但目前只有较短的描述直接显示在列表中。

### 第5步：包裹签名（可选）

第5步的说明还没有完全完成。您可能只想到第6步。

最后一步，这也是可选的，是签署您的存储库。这个加密过程向您的用户保证，他们正在下载的软件实际上来自您，而不是来自同一网络上拦截（和操纵）其流量的人。为此，我们首先将软件包文件的MD5和添加到Release，然后使用GnuPG（PGP的GNU版本）生成密钥并签署该文件，创建一个新的Release.gpg文件。

对于MD5和，我们添加了一个额外的字段调用“MD5Sum”，该字段在额外的行（以空格开头，因为它们不是字段）上有我们密钥文件的总和、大小（以字节为单位）和名称。请注意，我认为我们必须列出我们文件的未压缩版本和压缩版本。

```
MD5Sum: 07a4ca0b91734e0489874dc44bd55222 464 封43e92f4ec43f4c39d8f
```

不知何故，获得GPG密钥...（我不记得这个步骤了）。

现在我们需要从我们的发布文件创建我们的Release.gpg文件。此文件将由客户端首先下载，然后用于验证发布文件的有效性。

```
[root@desktop:/web/apt/xmpl]# gpg -abs -o Release.gpg 发布你需要一个
```

为了在客户端上验证此签名，他们必须安装我们的公钥。这是通过使用程序apt-key来完成的，apt-key是gpg的前端，为我们处理所有其他工作和论点。

```
iPhone:~ root# apt-key add saurik.pub好的iPhone: ~ root#
```

### 第6步：添加存储库

不幸的是，完全由于作者的懒惰，Cydia还没有一个好的编辑来添加来源。这太差劲了，会很匆忙地固定。与此同时，要添加我们刚刚创建的存储库，我们只需要在文件/etc/apt/sources.list中添加一行（在您创建之前可能不存在的），该行说明整个存储库的基本URL和此子分支的偏移量（通常为“/”，但在某些情况下可能更复杂）。

```
deb http://apt.saurik.com/xmpl/./
```

此示例行实际上是一个实时存储库，您可以根据本文中的说明随意测试或拆开和分析。在里面，您只会找到我们讨论过的两个文件，您可以安装包含MyProgram的软件包（它不起作用，如果您运行它，可能会导致问题）。

现在我们已经安装了存储库，我们可以继续测试它。为此，我们可以输入Cydia（当新源更新其软件包数据库时，它将自动接收新源），或者我们可以使用命令行工具“apt-get”。前者很明显，所以我会在这里介绍后者。首先，我们“apt-get update”以获取新的源信息，然后我们使用“apt-get install”来获取我们的软件包。

```
iPhone:~ root# apt-get 更新Ign http://apt.saurik.com./ Release.gpg
```

### 更多信息

此时，您应该完全准备好进入分发iPhone软件的世界。然而，应该指出的是，这只是冰山一角。我们现在构建的是一个“琐碎的存储库”，以及一个相当光秃秃的存储库。虽然对大多数用途来说绰绰有余，但一些用户的需求会更复杂。对于这些用户，我可以参考以下资源：

- [Debian存储库HOWTO](#)
- [Debian政策手册](#)
- [Debian系统](#)

最后，为了帮助整合我在{pre,post}{inst,rm}订购中找到的资源（包括各种操作中新旧软件包的所有参数），我正在编译以下列表：

- [随机邮件列表帖子](#)
- [Random Guy的网站](#)