**统计分析平台软件包发布流程**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** |
| 2014-02-21 | 0.01 | 初稿 | Ian Guo |
| 2014-02-21 | 0.0.2 | 增加post-install脚本描述 | Ian Guo |

目录

[1. 概述 3](#_Toc390344741)

[2. 软件包 3](#_Toc390344742)

[2.1 软件包命名格式 3](#_Toc390344743)

[2.2 软件包生成 3](#_Toc390344744)

[2.2.1 安装包 3](#_Toc390344745)

[2.2.2更新包 4](#_Toc390344746)

[3. 软件包发布流程 4](#_Toc390344747)

[3.1 下载代码 4](#_Toc390344748)

[3.2 增加版本号，写release notes 4](#_Toc390344749)

[3.3 编写post-install脚本（可选） 5](#_Toc390344750)

[3.4 生成软件包 5](#_Toc390344751)

[3.5 测试软件包 5](#_Toc390344752)

[3.6拷贝软件包到repository 5](#_Toc390344753)

[4. 其他问题 5](#_Toc390344754)

# 1. 概述

统计分析平台中有以下服务模块：stat-client, stat-server, db-server, config-server, stat-center。各个服务模块是独立的服务程序。其中stat-center负责对其他模块的监控，管理，包括自动更新。各服务模块的软件包存放在指定的目录中，stat-center对其统一管理。服务模块与stat-center建立连接后，实现自动下载，自动更新。所以，发布过程需要将服务模块制作成软件包，测试，放到stat-center指定的目录中。

为了同步线上服务和SVN中代码，以及规范服务模块的发布过程，避免低级失误，制定本流程。

# 2. 软件包

为了方便管理和维护各服务模块，在发布时利用makeself工具生成自解压的软件包。软件包分为两种：

* 安装包：包含完整的服务程序，运行后即可启动，正常使用。
* 更新包（升级包）：包含软件包中需要升级的部分，比如对bug的修复等，可使已安装的服务程序升级到最新。

## 2.1 软件包命名格式

软件包有统一的命名格式。

安装包格式：module-name-version-bz2.run，如stat-client-0.0.1-bz2.run

更新包格式：module-name-version-patch-bz2.run，如stat-server-0.0.2-patch-bz2.run

其中，module-name必须是以-分割，并且只有两段，比如：stat-center，db-server

version与linux库文件格式相同，必须以.分割，并且只有3段，分别为major.minor.patch，比如：0.0.2，0.2.1，1.3.1。

## 2.2 软件包生成

开发代码存放在SVN中，SVN：<http://svn.taomee.com/data-center/server/trunk>

运行svn co <http://svn.taomee.com/data-center/server/trunk> data-center下载代码。以下分别描述安装包和更新包的打包方法。

### 2.2.1 安装包

到相应服务模块目录中，运行**./build.sh dist**即可，安装包将生成并存放在update/release目录下。

### 2.2.2更新包

到相应服务模块目录中，运行./build.sh patch patch-file，patch-file列出需要包含的文件，以空格分割。比如**./build patch bin/stat-client.so bin/check-single**，将生成包含bin/stat-client.so和bin/check-single的更新包，更新包会生成在update/patch目录下。

**特别注意：**这种升级机制不支持跨版本升级，比如从0.0.2升级到0.0.5，如果0.0.3中包含一个文件，而在0.0.5更新包中不包括的话，升级可能出现问题。为了避免这种情况，务必在新版本更新包中包括旧版本更新包的全部内容。

# 3. 软件包发布流程

软件包发布流程如下：

1. 下载代码
2. 增加版本号，写release notes
3. 编写post-install脚本
4. 生成软件包
5. 测试软件包
6. 拷贝软件包到repository

以下详细叙述。

## 3.1 下载代码

运行svn update命令升级到指定版本（或到最新版本），比如svn update –r 1902。为了避免错误，强烈建议在一个独立的目录发布，不要和开发目录混合。发布目录只做发布，而开发目录只做开发，测试。

注意：发布的系统环境需要和目标运行系统的环境一致，如操作系统，操作系统版本，运行库等等。

## 3.2 增加版本号，写release notes

自动更新功能是以版本号为依据，所以在每次发布版本时，版本号需要增加。版本号记录在conf/version.conf中，打开该文件，修改版本号并保存，新版本号必须高于旧版本号。

release notes记录该次发布的相关信息，其内容在release-notes.txt中，格式如下：

第一行是发布日期和发布人

第二行是发布版本号

第三行是该次发布所对应SVN revision（即该版本SVN最后一次提交）

最后是Changelog，该次发布所做的修改。

新版本release notes在旧版本的顶端。

示例如下：

*2014-02-20 Ian Guo<ianguo@taomee.com>*

*version: 0.0.1*

*svn trunk revision: 1820*

*Changelog:*

*1. 完成自动更新功能，包括：检查更新，下载，安装。*

最后，将conf/version.conf和release-notes.txt提交到SVN中。每次发布打包脚本都会自动将上述两个文件加在软件包内，无需手工添加。

## 3.3 编写post-install脚本（可选）

在一些情况下，更新包安装完成后需要进行一些其他设置，比如修改配置等等，这些可以通过post-install脚本实现，这些脚本会在安装后被调用，如果脚本运行失败（比如exit 1），在安装过程终止，恢复原版本。如exit 0则安装成功。

在每个服务模块下有个post-install目录，post-install脚本需存放在此，脚本命名格式：post-install-version.sh，比如：post-install-0.2.0.sh，版本号和version.conf中配置的一样。

如果不需要post-install脚本，打包脚本会自动选择默认的post-install脚本，该脚本一定会执行成功。

## 3.4 生成软件包

生成软件包的方法如2.2节所述。如果软件包需运行在不同系统环境上，则需要在与之相同的环境上打包。

## 3.5 测试软件包

对生成的软件包进行测试，确保能在指定的系统环境上正确安装，运行。

## 3.6拷贝软件包到repository

repository是stat-center统一存放软件包的目录，其目录结构如下：os-name/os-version/module-name，比如debian系统，版本为6.0.2，stat-server的更新包应该存放在repository/debian/6.0.2/stat-server目录下。

至此，发布完成。相应的服务模块会自动升级。检查更新的频率默认是10分钟一次。

# 4. 其他问题

1. 刚发布的版本，发现有问题，怎么办？

首先，将已发布的软件包从repository中删除，然后，解决该问题或代码回退到能正确工作的版本，提交修改，再按发布流程发布新版本。

2. stat-server等服务端，新版本对旧版本的协议需要支持吗？

必须要，每次发布新版本必须支持旧版本协议。因为升级过程是自动的，有可能有的先更新有的后更新，后更新的会用老的协议，所以必须支持。

如果遇到无法解决支持旧版本协议时，需要手工操作，先将服务端停掉，升级客户端，然后升级服务端。