自动升级的设计思路与实现 - CJM恣肆 - 博客园

CJM恣肆关注 - 12 粉丝 - 67 +加关注

对于PC桌面应用程序而言，自动升级功能往往是必不可少的。而自动升级可以作为一个独立的C/S系统来开发，这样，就可以在不同的桌面应用中进行复用。为此我实现了一个可直接复用的自动升级系统。

目前主流的程序自动升级策略是，重新下载最新的安装包，然后重新安装整个客户端。这种方式虽然简单直观，但是缺陷也很明显。比如，即使整个客户端有100M，而本次更新仅仅只是修改了一个1k大小的dll，那也意味着要重新下载100M的全部内容。这对带宽是极大的浪费，而且延长了升级了时间，相应地也增加了客户茫然等待的时间。

在上述的场景中，自动升级时，我们能否只更新那个被修改了的1k的dll了？当然，使用OAUS自动升级系统可以轻松地做到这一点。OAUS自动升级系统可以对被分发的客户端程序中的每个文件进行版本管理，每次升级的基础单元不再是整个客户端程序，而是其中的单个文件。针对单个文件的更新，包括三种形式：

（1）文件被修改。

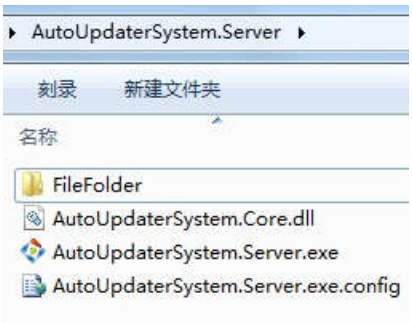
（2）文件被删除。

（3）新增加某个文件。

OAUS对这三种形式的文件更新都是支持的。每次自动升级，都可以更改N个文件、删除M个文件、新增加L个文件。

1.程序结构

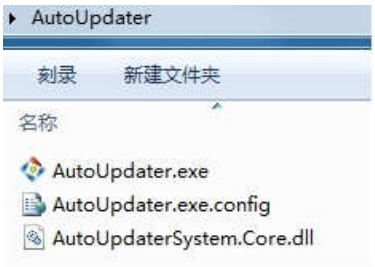
OAUS提供了可直接执行的服务端程序和客户端程序：AutoUpdaterSystem.Server.exe 和 AutoUpdater.exe。 OAUS服务端的目录结构如下所示：



OAUS的客户端与服务器之间通过TCP通信，可以在AutoUpdaterSystem.Server.exe.config配置文件中配置服务器通过哪个TCP端口提供自动升级服务。

FileFolder文件夹初始是空的，其用于部署被分发的程序的各个文件的最新版本。注意，其下的文件结构一定要与被分发的程序正常部署后的结构完全一致 -- 即相当于在FileFolder文件夹下部署一个被分发的程序。

OAUS客户端的目录结构如下：



可以在AutoUpdater.exe.config配置文件中配置OAUS服务器的IP、端口等信息，其内容如下所示：

```
<configuration>
  <appSettings>
    <!--服务器IP -->
    <add key="ServerIP" value="127.0.0.1"/>
    <!--服务器端口-->
    <add key="ServerPort" value="4530"/>
    <!--升级完成后，将被回调的可执行程序的名称-->
    <add key="CallbackExeName" value="Demo.exe"/>
    <!--主窗体的Title-->
    <add key="Title" value="文件更新"/>
  </appSettings>
</configuration>
```

请注意配置的CallbackExeName，其表示当升级完成之后，将被启动的分发程序的exe的名称。这个CallbackExeName配置的为什么是名称而不是路径了？这是因为使用和部署OAUS客户端时是有要求的：

（1）被分发的程序的可执行文件exe必须位于部署目录的根目录。

（2）OAUS的客户端（即整个AutoUpdater文件夹)也必须位于这个根目录。

如此，AutoUpdater就知道分发程序的exe相对自己的路径，如此就可以确定分发程序的exe的绝对路径，所以就可以在升级完成后启动目标exe了。另外，根据上述的两个约定，再结合前面讲到的服务端的FileFolder文件夹的结构约定，当服务端更新一个文件时，AutoUpdater便可以确定该文件在客户端机器上的绝对路径了。

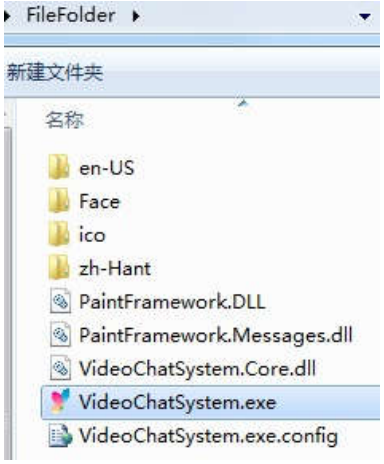
2.自动升级流程

下面我们就详细讲讲如何使用OAUS来构建自动升级系统，大概的步骤如下。

（1）运行OAUS服务端。

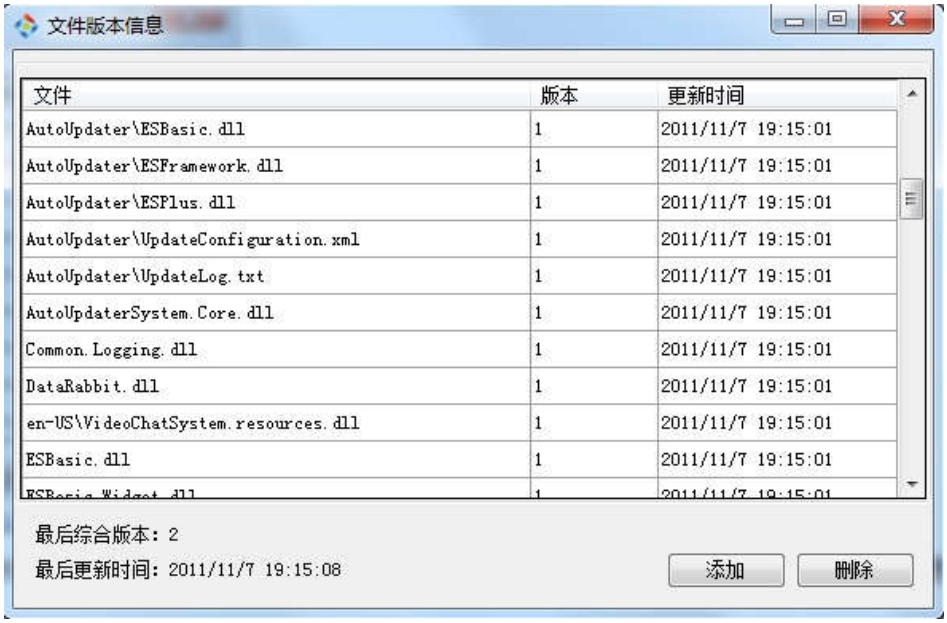
服务端主界面将显示所有正在自动升级的OAUS客户端信息。

（2）将被分发的客户端程序的所有内容放到OAUS服务端的FileFolder文件夹下，其结构与客户端程序正常部署后的结构要完全一致。我们以部署VideoChatSystem为例。



（3）使用OAUS服务端为被分发的客户端程序的每个文件生成默认版本号，并创建版本信息配置文件UpdateConfiguration.xml。这个配置文件也将被客户端使用。

点击服务端【工具】菜单栏下的【版本管理】子菜单，将弹出用于管理各个文件版本的【文件版本信息】窗体。



双击列表中的任意一行，可以修改其对应文件的版本的值（float类型的数值）。注意，此列表中的版本信息与文件的真实版本属性（比如dll的版本属性X.X.X.X）可以是没有任何联系的，列表中版本的值只是用于标记文件是否被修改，所以，文件每被修改一次，其列表中对应的版本的值就应该有所增大。

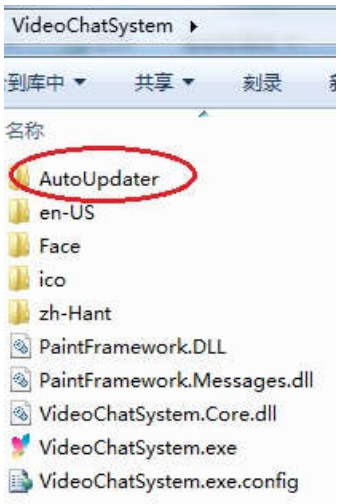
当关闭【文件版本信息】窗体时，只要有某个文件版本变化，则“最后综合版本”的值（int类型）会递增1。通过比较OAUS的客户端保存的“最后综合版本”的值与OAUS的服务端最新的“最后综合版本”的值，就可以快速地识别客户端是否已经是最新版本了。

另外，初次打开这个窗口时，将在OAUS服务端的目录下，自动生成一个版本信息配置文件UpdateConfiguration.xml。而且，每当通过该窗体来设置某个文件的新版本时，UpdateConfiguration.xml会自动同步更新。

（4）将UpdateConfiguration.xml添加到OAUS的客户端程序（即上述的AutoUpdater的文件夹）中。

（5）在创建被分发的客户端的安装程序时，将OAUS的客户端（即AutoUpdater的文件夹）也打包进去，并且像前面说的一样，要将其直接部署在运行目录（BaseDirectory）下（与分发的exe同一目录）。

如此，准备工作就完成了，当客户端通过安装包安装好了VideoChatSystem之后，其目录结构像下面这样：



（6）当我们有新的版本要发布时，比如要更新某个文件（因为文件被修改），那么可以这样做：

a.将修改后的文件拷贝到OAUS服务端的FileFolder文件夹下的正确位置（或覆盖旧的文件）。

b.在OAUS服务端打开【文件版本信息】窗体，双击被修改文件所对应的Row，在弹出的窗体上修改对应文件的版本号，将版本号的数值增加。（如果是删除旧文件或添加新文件，此处也可进行相应的操作）

（7）如此，当客户端再启动AutoUpdater.exe时，就会自动升级，更新那些发生变化的文件。 以下是AutoUpdater.exe运行起来后的截图。

（8）当升级完成后，将启动前述的OAUS客户端配置文件中配置的回调exe。（在本例中就是VideoChatSystem.exe）

（9）OAUS客户端会在日志文件UpdateLog.txt（位于AutoUpdater的文件夹下，在OAUS客户端首次运行时自动生成该文件）中，记录每次自动升级的情况。

3.何时启动自动升级客户端?

假设某个系统是下载客户端形式的，那么客户端该如何知道是否有新版本了？然后又该何时启动AutoUpdater.exe了？

我们的经验是这样的：客户端登录成功之后，从服务器获取“最后综合版本”的值，然后与本地的“最后综合版本”的值相比较，如果本地的值较小，则表示客户端需要更新。这个过程可以这样做到：

（1）当在OAUS服务端的FileFolder文件夹下放置了新的文件，并通过【文件版本信息】窗体正确的更新了版本号，在关闭【文件版本信息】窗体时，“最后综合版本”的值会自动加1。

（2）系统客户端可以通过调用AutoUpdater.VersionHelper类的静态方法HasNewVersion()来判断是否有新版本。

（3）如果HasNewVersion方法返回true，则通常有两种模式：由用户选择是否升级，或者是强制升级。

一般而言，如果最新客户端程序与老版本兼容，不升级也影响不大，则可以交由用户决定是否升级；如果最新客户端程序不兼容老版本，或者是有重大更新，则将启动强制升级。如果流程要进入启动升级，那么只要启动AutoUpdater的文件夹下AutoUpdater.exe就可以了。要注意的是，启动AutoUpdater.exe进程后，要退出当前的客户端进程，否则，有些文件会因为无法被覆盖而导致更新失败。代码大致如下所示：

```
if (VersionHelper.HasNewVersion(oausServerIP,oausServerPort))
  {
       string updateExePath = AppDomain.CurrentDomain.BaseDirectory + "AutoUpdater\\AutoUpdater.exe";
       System.Diagnostics.Process myProcess = System.Diagnostics.Process.Start(updateExePath);
       ......//退出当前进程
  }
```

三.相关下载

源码