

自动更新库使用手册

版本：3.0.5405.37479，更新日期：2014 年 10 月 19 日

本类库+工具用于快速实现一个简单的自动更新程序，可以简单地完成小型应用程序的快速更新。

整个工具分成两部分，一部分是供用户使用的类库，一部分是用于生成更新包的工具。

功能支持：

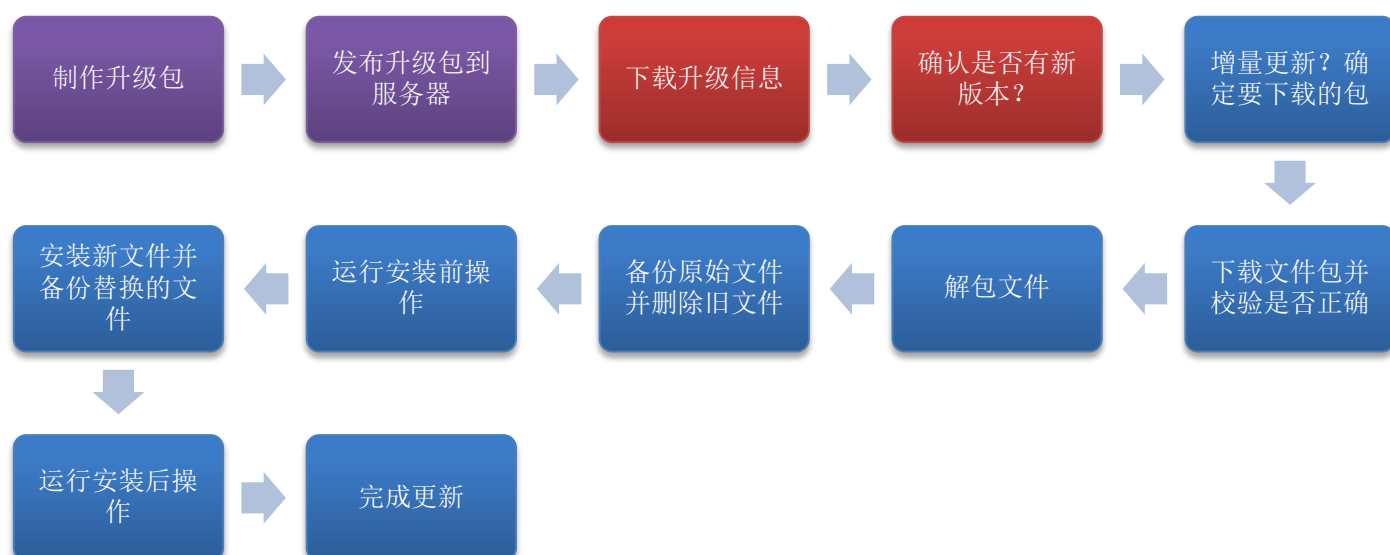
- 多种调用方式，支持引用自动升级客户端进行调用，或使用命令行进行调用；
- 调用简单，最简单的情况下一行代码完成自动更新；
- 全自动化的流程，自动下载，自动校验升级包，自动结束正在运行的程序；
- 支持增量更新；
- 安装的备份功能，如果更新时发生错误，能还原到安装更新前的状态；

系统需求：

- 客户端要求至少 V2.0 Framework 的支持，兼容 V2.0-V4.0；
- 升级包制作工具要求至少 V3.5 Framework 的支持，兼容 V3.5-V4.0；

1. 升级流程

自动升级库整体的工作流程如下：



图中，紫色的步骤代表服务器端发布更新的操作；红色的步骤表示在程序中检测更新时所完成的操作；蓝色的步骤则是正式更新的操作。

一般来说，在检测到更新后，自动更新客户端会在系统的临时目录中创建一个随机目录，将自己复制到其中，然后启动其中的副本，进行正式的自动更新。要进行这步操作的原因是：

- 需要将自动更新客户端脱离原目录，防止文件被占用无法更新；
- 保存下载的包文件和解压出来的临时文件；
- 备份在安装过程中删除或替换的文件，以便于出错时还原。

2. 制作升级包

要进行更新，第一步就是制作升级包并发布到服务器上。最新版（2.0.0.0）升级包制作工具可以制作出兼容老版本升级库升级的升级文件。但是，在启用增量更新的情况下，会生成很多的包文件（依据程序文件的多少和文件更新方式的不同而不同）。因此，建议每个软件的升级包都单独按目录放置。

启动包制作工具，主界面如下：

升级包创建工具 - BY iFish, Version 3.0.5405.37035

基本信息更新说明服务器选项更新选项1更新选项2文件配置命令行生成关于...

应用程序名

当前版本

绑定

发布地址

新程序目录

浏览

升级包路径

浏览

更新前执行

<未选择>

☐ 隐藏

更新后执行

<未选择>

☐ 隐藏

执行时间限制

0

(秒, 0为不限制)

保存升级项目

打开升级项目

创建升级包

- 主界面的功能区分为七块：
- **基本信息**：设置应用程序的基本信息，版本信息，更新说明，和**保存升级包的目录**。注意，老版本中这里是升级包文件（*.zip）的路径，在新版中，因为增量更新的引入，已经修改为保存升级包的目录；
 - **更新说明**：设置在更新提示框里的说明信息
 - **服务器选项**：设置部分更新时的服务器通知选项
 - **更新选项 1**：设置升级包生成的选项，如最低支持升级的版本、升级包密码，老版本应用程序删除的方式，以及兼容老版本升级的兼容包选项；
 - **更新选项 2**：设置更多的选项

- **文件配置**: 用于设置增量更新时文件更新模式的选项;
- **命令行生成**: 生成用于命令行调用模式时, 需要使用的命令行参数;
- **关于**: 查看关于工具的相关信息

『打开升级项目』功能会要求你选择一个已经保存的项目文件 “*.auproj”, 工具会自动加载所有信息以及文件升级选项配置, 只需要再选择一个新版本软件的目录即可。或者, 您可以直接将升级信息文件直接拖放到主界面上来打开。

2.1 基本信息设置

基本信息设置用于设置应用程序的基本信息。详细介绍如下:

选项	说明
应用程序名	设置当前升级包相对应的软件名。
当前版本	设置当前服务器上最新的发布版本, 这是升级的依据。
绑定	将版本信息绑定到一个 DLL 或 EXE 文件。绑定后, 不可填写版本, 当构建升级包的时候, 会自动从对应的文件中读取版本信息
发布地址	用于填写软件主页。可以不填, 这里填写与否, 将会影响到最终软件提示界面上的主页链接。
新程序目录	这里选择最新版本软件的根目录, 打包的新文件将会从这里提取。
升级包路径	设置用于保存 升级包路径的目录 。注意, 老版本中这里是升级包文件 (*.zip) 的路径, 在新版中, 因为增量更新的引入, 已经修改为保存升级包的目录。
更新前执行	在升级包下载并解压缩完成、即将开始安装前, 进行执行的程序。更新文件的安装操作将会在执行完成后继续; 更新前执行的软件将会在升级的临时目录执行, 可用相关的环境变量请参见 4.1 可用的环境变量 ; 选择文件框后面的文本框用于填写命令行参数; 请注意 4.2 注意事项 中提到的内容。
更新后执行	在更新文件安装后, 进行执行的程序。此时, 更新已经成功, 自动升级程序不会等待进程退出便提示成功; 更新后执行的软件将会在软件的原目录执行, 可用相关的环境变量请参见 4.1 可用的环境变量 ; 选择文件框后面的文本框用于填写命令行参数; 请注意 4.2 注意事项 中提到的内容。
执行时间限制	用于限制 更新前执行 的进程允许执行的时间, 为 0 时永远等待; 不为 0 时, 如果启动的进程在指定的时间 (秒为单位) 后没有退出, 自动升级程序将会提示是否强行中止进程;
更新说明	用于填写当前最新版本的升级说明。
隐藏	选择此选项的话, 将会隐藏对应进程执行时的窗口 (如果有)

2.2 更新说明



升级包创建工具 - BY iFish, Version 3.0.0.0 Build 20140705

基本信息 更新说明 服务器选项 更新选项 文件配置 命令行生成 关于...

在2.2.0.0版本之后的库中，同时支持纯文字、内嵌网页以及RTF样式的更新说明。默认情况下，内嵌网页如果填写，则优先显示内嵌网页；否则优先显示RTF格式文本，最后再显示纯文字版。

考虑到之前版本的兼容性，如果您有发布过使用老版本升级库的应用程序，建议始终填写纯文字版，在此基础上再填写网页链接或选择RTF文件。注意，RTF文件可能导致最终的说明文件体积很大，请尽量使用压缩版XML信息文件。

纯文字版 [绑定到文件...](#)

更新说明网址

RTF格式说明文件 浏览 编辑

保存升级项目 打开升级项目 创建升级包

各项说明参见上图。“绑定到文件”和前面的绑定版本信息类似，也是用于自动化处理的手段。

2.2 服务器选项



服务器选项是用于统计用途的。其中的**通知地址**将会在每次正式更新之前向指定的服务器地址发送请求，以便于做更新统计。

2.3 更新选项 1 设置

更新选项页界面如下：

升级包创建工具 - BY iFish, Version 3.0.5405.37035

基本信息更新说明服务器选项更新选项1更新选项2文件配置命令行生成关于...

支持更新的最低版本升级文件包密码☒ 使用增量更新

☒ 生成压缩版的升级信息文件。这将有助于在启用增量更新时大幅减小升级信息文件大小。请使用“update_c.xml”作为升级信息文件名。建议选中『创建兼容的升级信息包』，否则2.0.0.0之前的自动升级客户端将无法执行升级。

☒ 创建兼容的升级信息包。如果不选择此选项且有文件使用增量更新方式发布，在2.0.0.0版本之前（不支持增量更新的自动更新客户端）将无法使用此处的安装包进行更新。

默认文件检测更新模式☐ 忽略更新☐ 始终更新☐ 不存在时更新☒ 比较版本更新 MD5

更新时删除原始文件

仅覆盖，不主动删除

保存升级项目打开升级项目

创建升级包

选项解释……看表吧亲~

选项	说明
支持的最低版本	指定升级包可以升级的最低客户端版本。低于此版本的客户端将会提示版本过旧要求去主页手动更新，而不是执行更新。
升级包密码	指定生成升级包使用的密码。留空则不使用密码。
使用增量更新	设置是否启用增量更新。如果不使用增量更新，则生成的是和老版本升级程序完全兼容的升级包。
压缩版升级信息文件	设置是否生成压缩版本的升级信息文件（指的是 xml 文件）。在启用增量更新的情况下，文件稍微多点时，将会导致最终的 xml 文件很大。如果启用此选项，工具将会对其进行压缩，生成压缩版的 xml 文件，一般会缩小 50%-80%的体积。但是由于老版本的升级库不支持这种格式的 xml 信息文件，因此这个选项将会生成新的 xml 文件（默认为 update_c.xml ）。如果你希望使用压缩版的信息文件来升级，请在设置升级路径的时候使用 update_c.xml 。注意，老版本的升级库，请继续使用 update.xml 。
兼容的升级信息包	设置是否创建兼容的升级信息包。这项设置主要针对增量更新设置的。在不使用增量更新（所有的文件都是 始终更新 时），是否勾选此选项没有影响，因为生成的文件包都是可以同时适用于老版本升级程序的。但是建议此时依然勾选此选项，因为如果此选项不勾选，在启用“压缩版升级信息文件”选项时，老版本兼容的升级信息文件 update.xml 将会被自动删除，不再保留，这会导致虽然升级包兼容，但是老版本的应用软件依然无法使用这里的升级包进行升级。
删除原始文件	默认情况下，自动升级程序不会对旧应用软件目录做出任何修改，而是直接复制新文件（覆盖或新加）。但是在这里，你可以设置为在安装前强制删除原目录的文件。删除模式有两种： 1. 只删除指定的目录和文件；

	2. 删除除指定的目录和文件以外所有其余的目录和文件；
删除或保留的文件	<p>设置要保留或要删除的文件和目录。是删除指定的还是删除指定以外的要看在上一个选项中的删除方式。</p> <p>这里的文件一行是一条规则，规则使用正则表达式进行判断。</p> <p>判断的路径是相对目录。比如，软件安装目录是 <code>C:\Soft\FishTools\</code>，那么程序目录下的文件 <code>C:\Soft\FishTools\Bin\Main.exe</code> 的相对路径将是 <code>Bin\Main.exe</code>，规则“<code>Bin\\Main.exe</code>”将会匹配它。</p>
默认文件检测更新方式	当源目录中出现了未显式指定更新方式的文件时，将会使用此处的方式进行检测是否需要更新。

2.4 更新选项 2

更新选项 2 是在 3.0 版的客户端中才引入的选项卡。这些选项将原本需要从客户端才能设置的选项提升到打包的级别。

基本信息更新说明服务器选项更新选项1更新选项2文件配置命令行生成关于...

☐ 不提示直接自动启动升级

如果启用此选项，则当找到更新时，用户不会看到是否更新的确认提示框，也不会提示更新内容，直接启动更新程序。

☐ 更新文件时自动结束进程

如果启用此选项，则当安装新文件的时候，如果需要结束有关进程，将不会提示用户关闭，而是直接强行关闭。

☒ 更新文件时，同时结束软件目录下其它的进程

如果启用此选项，则当安装新文件的时候，如果需要结束有关进程，将不会提示用户关闭，而是直接强行关闭。

☐ 如果不安装此更新，则强制退出软件

指示当前的更新是必须更新的内容，而无法选择是否更新。这个选项开启后，将会强行开启“不提示便自动启动升级”。当用户选择不升级的时候，进程将会被强行退出。

☐ 启动更新后，自动退出当前进程

如果启用此选项，当更新程序启动的时候，当前的进程将会被自动退出。

☐ 如果检查更新出现错误，则按照有更新处理

设置当出现错误的时候，是否按照有更新但是未更新处理。这个选项影响设置必须强制更新的选项。如果检测更新遇到错误，此选项未开启时，则按照“未找到更新”处理；如果此选项已开启，则按照“有更新但是没有更新”处理，会强制退出软件。

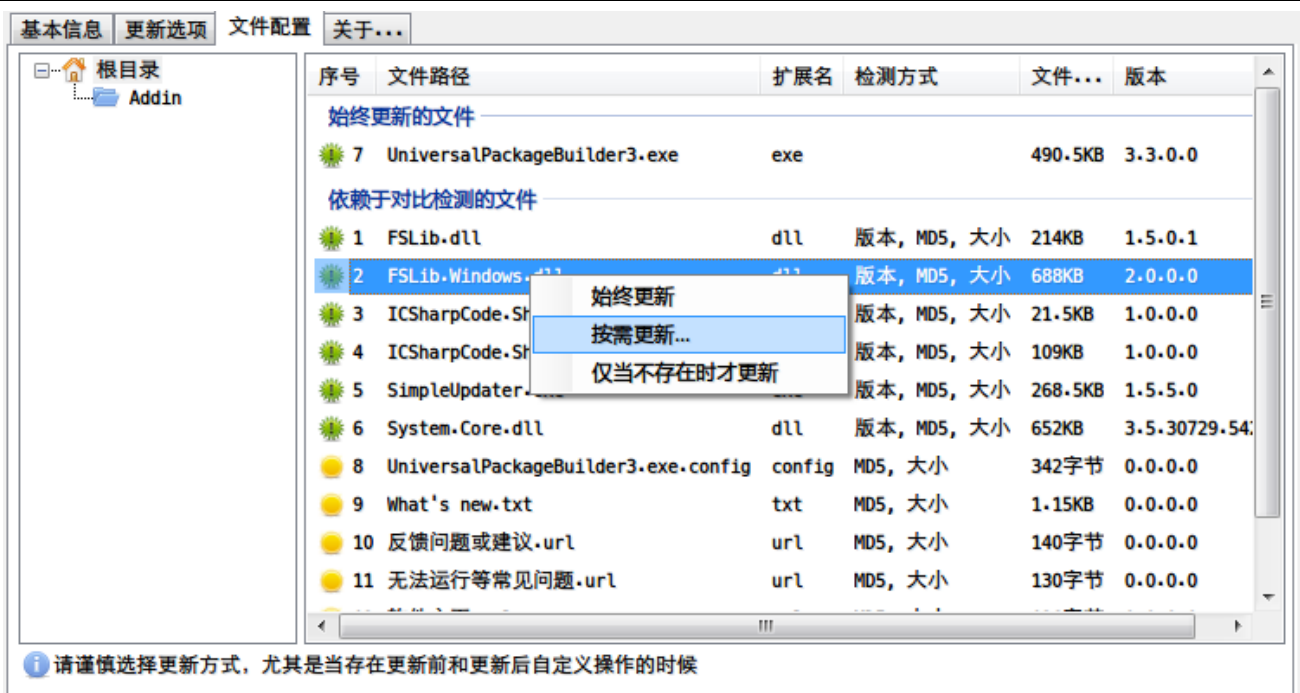
☒ 不显示版本更新内容而直接升级时，在开始前依然给用户一个提示信息（仅通知开始升级）

☐ 针对所有自动更新操作均要求管理员权限

更新程序通常会自动检测是否有权限对当前目录操作并申请管理员权限。如果勾选此选项，则当开启UAC时，升级程序会全部请求管理员权限。

由于选项卡上已经增加了比较直接完善的文字解释，此处不多做解释。

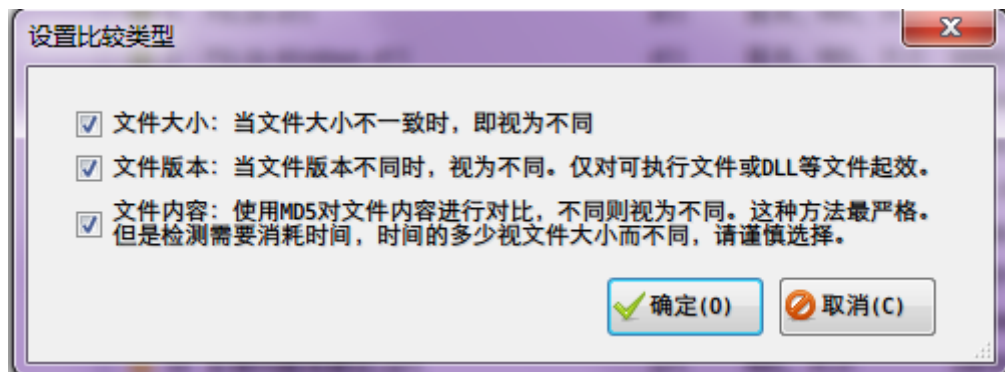
2.5 文件配置



文件配置用于配置具体的文件在更新时的动作，这将决定在进行增量更新时，对应的文件包是否下载。通过增量更新，在更新幅度比较小时，可以大幅减少版本更新所需要下载的数据量。

右侧的文件列表中，更新选项是以分组的形式表现出来的。『始终更新的文件』表示这部分文件不进行任何对比，不管当前版本和内容是否匹配，都一律下载更新并覆盖；『依赖于对比检测的文件』表示按照对应的检测方法来确定客户端文件是否和服务端相匹配，如果不匹配则进行更新；『仅当不存在时才更新』表示，只要客户端存在这个文件，就不更新（不管内容是否相同）。

设置比较类型时，允许设置三种比较类型，如下图所示。

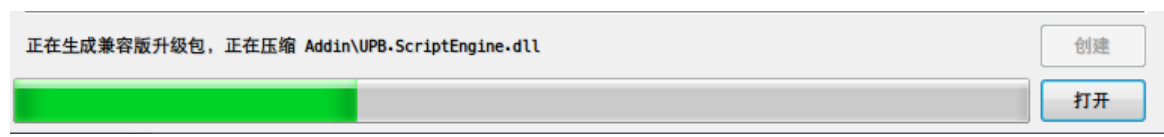


文件版本仅针对可执行文件和动态链接库等具有版本信息的文件才起效。不具备版本信息的常规文件，版本一律识别为“0.0.0.0”，如果对这些文件选择了『文件版本』，工具将会进行警告。

文件内容比较是使用 MD5 对文件的每个字节进行比较的方法。这个方法最为准确，可以精确确认两个文件是否一样。缺点是计算 MD5 时需要时间（视文件大小和数量而不同），因此请酌情考虑。

注意：在设置了安装前操作时，请确认对应操作所需要的文件您是全部设置为『始终更新』的。因为安装前操作被执行时，进程所在目录位于升级包解压的临时目录，因此如果对应的文件被设置为增量更新却没有下载，会导致执行失败。

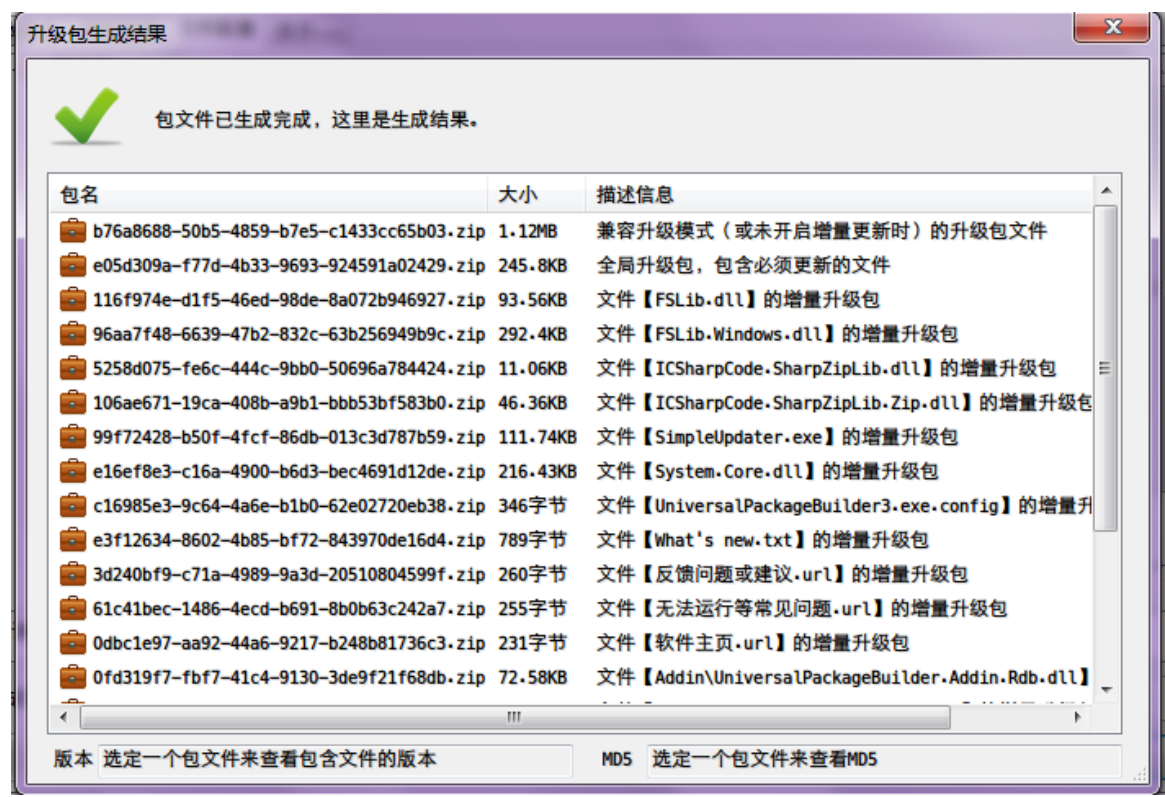
2.5 创建升级包



设置全部完成后，点击『创建』即可完成创建工作。

一般创建完成后，打包工具会反馈一个创建的最终结果，包括创建了哪些文件、每个文件的大小、以及对应的注释，如下图所示。

在启用增量更新的情况下，会有一个或多个 zip 压缩包被生成，以及 1 个或多个 xml 信息文件。



3. 启动自动更新

在继续前，我们需要了解几个概念：

- 下载模板 URL。升级库支持以下几种升级地址：**WEB 下载** (<http://>, <https://>)；**FTP 地址** (<ftp://>)；**本地地址** (如 `c:\xx\xx`)；**局域网共享地址** (如 `\\update-server\soft\update\update.xml`)。传递给升级库的地址我们称为模板地址（模板 URL），将会使用这个模板地址生成最终的下载地址。在模板地址中，我们使用“{0}”作为文件名的占位符。之所以要使用占位符，是因为在整个升级过程中，我们需要下载一个或更多的文件，因此，将文件名作为占位符后，可以直接使用模板 URL 来合成所有的升级地址。
例如，我们设置模板 URL 为 <http://update.mydomain.com/test-soft/{0}>，那么最终生成下载文件“update.xml”、“update.zip”时，升级库将会把地址中的“{0}”分别替换为文件名，并使用此地址来下载。
- 更新文件名。更新文件名一般默认为“update.xml”（兼容模式的升级信息）或“update_c.xml”（经过压缩

的升级信息文件)。升级库使用此文件名和上面的模板 URL 生成最终的升级信息地址。

- 当前版本。用于比较当前版本和服务端版本之用。默认情况下，当前版本使用当前入口程序集（也就是正在执行的程序）的版本号作为版本号。在命令行调用的情况下（或特殊的原因时）我们需要指定自己的版本号时，可以通过上下文环境（程序集引用）或命令行参数（命令行调用时）进行指定。详情请参考后面的章节。
- 应用程序当前目录。当前目录是更新的基础目录，具体的概念请参考[相对路径](#)中的介绍。默认情况下，当前目录是正在执行的进程（程序集引用）或升级库所在目录（命令行调用）的地址作为当前目录。需要修改时（比如当前的程序实际上位于应用软件目录的 Bin 目录下时），可以通过上下文环境（程序集引用）或命令行参数（命令行调用时）进行指定。详情请参考后面的章节。

3.1 程序集引用（推荐）

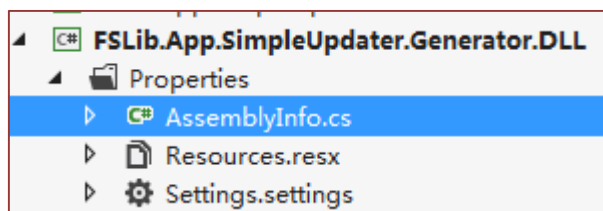
3.1.1 添加引用

要使用这种方式进行自动更新，第一步是添加程序集引用。请添加此程序集引用：

客户端\DLL\SimpleUpdater.dll。

3.1.2 添加升级标记

添加完成后，需要标记升级地址。升级地址可以通过程序集属性声明，也可以在调用升级函数的时候通过构造函数或静态函数传入。如果你希望通过属性来声明，那么请在 VS 中打开最终的可执行文件项目（也就是输出为 exe 文件的项目）中“**Properties\AssemblyInfo.cs**”（C#中，VB.net 中对应“**Properties\AssemblyInfo.vb**”）文件：



在文件中，添加一行这样的声明：

```
[assembly:FSLib.App.SimpleUpdater.Updatable2("http://www.yourdomain.com/update/test-soft/{0}",  
"update_c.xml")]
```

这个属性生命，第一个参数表示模板 URL，第二个参数表示升级信息文件名。
如果不希望通过这个升级属性来标记，可以跳过，直接进行第三步。

3.1.3 调用升级代码

在大多数情况下，我们只需要一个升级实例即可。那么此时，我们可以直接调用静态函数即可。
静态函数的调用入口共有三个重载，除此以外，亦可以通过实例方法进行调用。参见如下代码：

代码

```
//1.直接进行更新。这个入口不需要任何参数，因此请确认在调用前已经通过代码或属性初始化过升级地址。  
//否则，会抛出异常  
Updater.CheckUpdateSimple();  
  
//2.调用一个参数的重载。这个重载是作为兼容模式而引入的，请尽量避免使用。  
//接受的参数为一个XML升级信息的地址。这种方式只支持HTTP方式，而且要求XML文件名位于地址的最后面  
//否则执行会出异常  
Updater.CheckUpdateSimple("http://www.yourdomain.com/soft/update/update.xml");  
  
//3.调用两个参数的重载。这俩参数和属性声明是对应的。使用此重载时，不需要用属性进行标记地址。  
Updater.CheckUpdateSimple("http://www.yourdomain.com/soft/update/{0}", "update_c.xml");  
  
//4.通过实例对象进行手动更新  
var updater = Updater.Instance;  
//这里可以设置一些Updater的属性  
//然后手动调用检查更新  
updater.BeginCheckUpdateInProcess();
```

注意：以上代码演示了四种更新方法，但在实际的操作中，只需要其中一种就行了。推荐使用第三种方法。

3.1.4 事件捕捉

升级库有很多事件，基本上都是实例事件。在绝大部分情况下，我们可以通过 `Updater.Instance` 获得当前的唯一实例对象，从而简化代码。通常情况下经常用到的事件如下：

事件名	描述
Error	当操作发生错误时引发。引发此事件后，可以通过 <code>Updater</code> 对象的 <code>Context.Exception</code> 属性来获得发生的错误。
MinmumVersionRequired	当前应用程序的版本不符合升级文件中定义的最低支持版本。
NoUpdatesFound	没有更新的版本
StartExternalUpdate	正在启动外部更新进程进行正式更新
UpdatesFound	已经找到新版本

在调用静态函数进行检查更新时，默认情况下将会启用内置的提示对话框。这将有利于实现全傻瓜化地更新，在您的应用程序中不需要再去处理任何事件。内置的提示对话框和上面的常见事件对应关系如下：

事件名	对应内置对话框
Error	(无)
MinmumVersionRequired	提示用户当前版本过低，并提供主页链接供手动更新。
NoUpdatesFound	(无)
StartExternalUpdate	(无)
UpdatesFound	提示用户找到新版本，确认是否更新。

如果您希望禁用这些内置的对话框，可以将 `Updater.Instance.Context.EnableEmbedDialog` 属性设置为 `false` 即可。

在您禁用后，检测到更新时，升级库将不会有任何动作，只有您自己手动捕捉这些事件并进行更新。确认要进行更新的话，请调用 `Updater.StartExternalUpdater()` 方法来启动最终的更新。

注意：如果您需要禁用内置的对话框，请勿使用 `CheckUpdateSimple` 的三个静态重载函数来进行更新，这三个函数将会默认恢复内置的提示对话框。

3.1.5 可设置的参数

自动升级库的实例对象拥有 `Context` 属性，可以获得当前更新的上下文环境和选项。具体可设置或可访问的属性，请参考下表。

属性名	访问性	说明
<code>IsInUpdateMode</code>	只读	获得当前是否处于正式更新（安装更新）阶段。
<code>AutoEndProcessesWithinAppDir</code>	读/写	获得或设置更新时是否要求结束所有位于应用程序所在目录下的进程，以便于正确更新。
<code>ExternalProcessID</code>	只读	获得进行更新时所有需要进行结束的进程 ID 列表。
<code>ExternalProcessName</code>	只读	获得进行更新时所有需要进行结束的进程名列表。
<code>Exception</code>	只读	获得在进行操作时发生的错误
<code>UpdateDownloadUrl</code>	读/写	获得或设置用于更新的模板地址
<code>UpdateInfoFileName</code>	读/写	获得或设置更新时使用的 XML 文件名
<code>UpdateAttribute</code>	读/写	获得或设置提供更新地址信息的属性对象。一般来说不需要设置这个属性。
<code>ApplicationDirectory</code>	读/写	获得或设置当前的应用程序目录。默认为当前进程所在目录。可以设置为绝对或相对路径，如果设置为相对路径，那么将会使用当前的路径和设置的相对路径进行合并得到新的路径。
<code>UpdateInfoFileUrl</code>	只读	获得用于下载更新信息文件的地址
<code>CurrentVersion</code>	读/写	获得或设置当前的应用程序版本。默认为当前进程的主文件的版本号，如果需要手动指定，则设置此版本号。
<code>UpdateInfoTextContent</code>	只读	获得当前更新信息序列化后的文本信息
<code>UpdateInfo</code>	只读	获得当前更新信息的实例对象
<code>LatestVersion</code>	只读	获得当前检查到的最新版本
<code>EnableEmbedDialog</code>	读/写	获得或设置是否启用内置的对话框
<code>IsInUpdating</code>	只读	获得或设置是否正在进行更新中
<code>MultipleDownloadCount</code>	读/写	获得或设置同时下载的文件数，默认为 3。
<code>MaxiumRetryDownloadCount</code>	读/写	获得或设置重试的最大次数，默认为 3。
<code>UpdateTempRoot</code>	只读	获得当前更新的临时目录
<code>UpdateInfoFilePath</code>	只读	获得当前更新信息文件保存的路径
<code>UpdatePackageListPath</code>	只读	获得当前要下载的包文件信息保存的路径
<code>UpdatePackagePath</code>	只读	获得当前下载的包文件目录
<code>UpdateNewFilePath</code>	只读	获得当前下载解包后的新文件路径
<code>UpdateRollbackPath</code>	只读	获得当前更新过程中备份文件的路径
<code>IsUpdateInfoDownloaded</code>	只读	获得一个值，表示当前的自动升级信息是否已经下载完全
<code>NetworkCredential</code>	读/写	获得或设置服务器用户名密码
<code>ProxyAddress</code>	读/写	获得或设置用于下载的代理服务器地址

HasUpdate	只读	获得是否找到更新
CurrentVersionTooLow	只读	获得表示是否当前版本过低而无法升级
LogFile	读/写	获得或设置日志文件名。可以是相对或绝对路径，如果设置的是相对路径，那么保存的日志文件将位于系统临时目录下。
ForceUpdate	读/写	获得或设置是否不经提示便自动更新
AutoKillProcesses	读/写	获得或设置是否在更新时自动结束进程

3.2 命令行调用

命令行调用模式是在 2.0.0.0 版本的升级库中新增的功能。通过这个模式，可以实现在完全不对已有的软件进行任何结构上修改的情况下完成对软件的更新。因此，也完全可以实现对非.Net 软件以及各种莫名的文件的更新……

3.2.1 添加升级文件

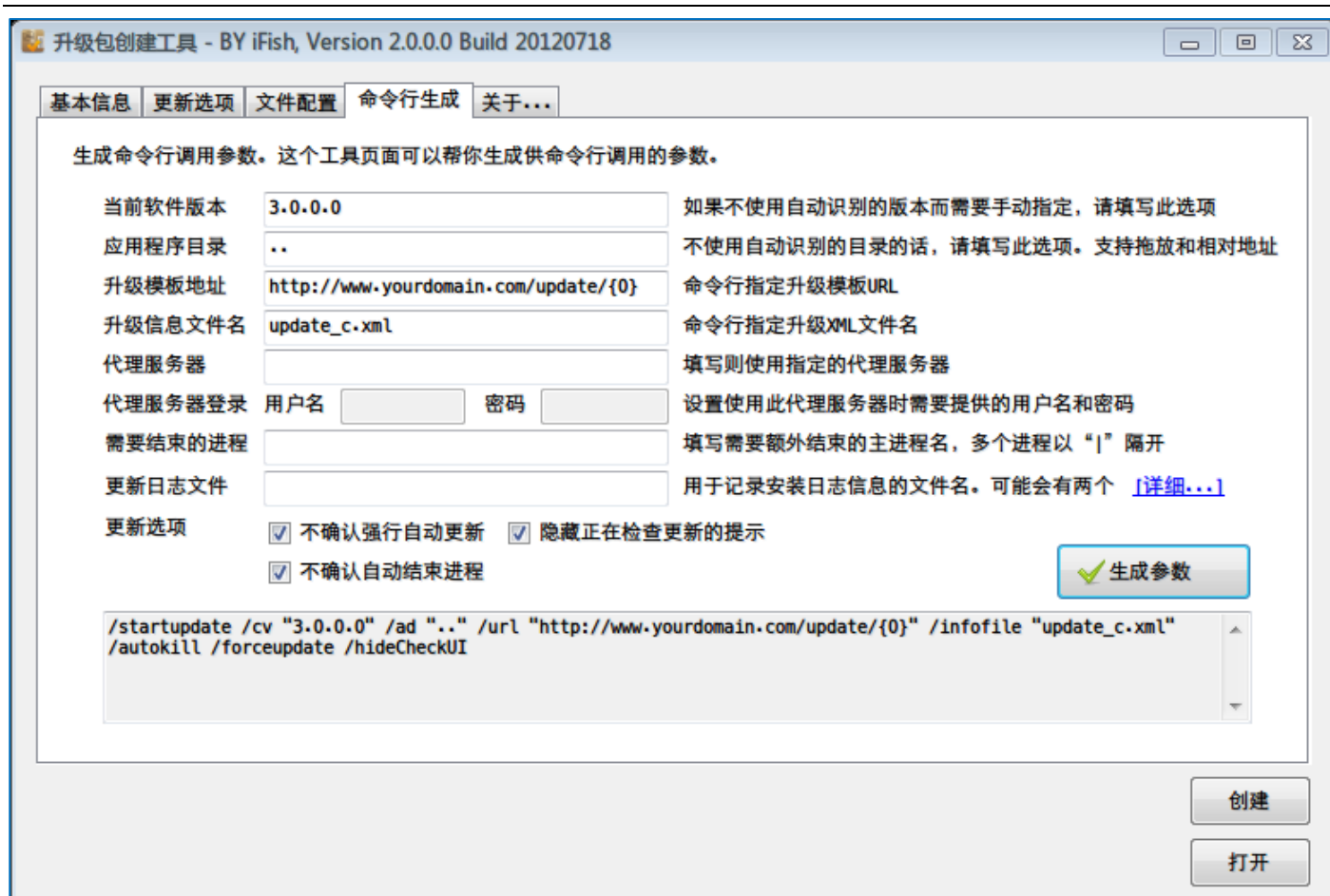
复制以下文件到您的软件目录：

- 客户端\EXE\SimpleUpdater.exe
- 客户端\EXE\SimpleUpdater.exe.config

其中，SimpleUpdater.exe.config 用作 Framework 的版本兼容文件，有了此文件的支持，升级的客户端程序可以在 2.0-4.0 的 Framework 下运行。若不想要此文件，请自行编译源代码，并在编译前修改项目的目标平台版本。

3.2.2 使用工具生成命令行

打包工具中的『命令行生成』标签页用于生成命令行。



若您已经对命令行参数较为熟悉，那么也可以直接填写，而不使用工具生成。界面中的各项参数的具体意义，请参见本手册之前相关章节的说明。

3.2.3 调用

准备好可执行程序 and 参数后，直接使用指定的参数来调用 **SimpleUpdater.exe** 即可。

3.2.4 提醒

在客户端调用下，将会强制使用内置的对话框，同时，任何内部的属性和上下文环境，以及事件，您将都无法进行访问。

4. 命令行构建更新项目

最新版更新项目创建工具支持用命令行方式创建。在这种模式下，推荐在前述的绑定中将指定的版本信息和更新说明绑定到特定文件中，之后使用命令行即可创建更新项目。

打包工具支持的命令行参数如下所示。

自动更新包生成工具.exe [/build] [/noui] [/miniui] 项目文件

开关解释如下：

1. `/build` 此开关将会使项目打开后就开始立刻创建目标更新包
2. `/noui` 此开关将会导致创建时不显示独立的对话框，而显示为控制台消息
3. `/miniui` 此开关将会导致创建时不显示主界面，仅显示进度条界面
4. 项目文件是必须的，否则所有开关都会无效。
5. `/noui` 和 `/miniui` 只能存在一个，或一个都没有。都不指定时，显示为主界面。

5. 相关问题

4.1 可用的环境变量

在安装前执行、安装后执行的操作中，2.0.0.0 开始的自动更新库更新了执行代码，强化了信息访问。在执行的文件是可执行文件（EXE/COM），或批处理文件（CMD/BAT）时，更新库会将当前所有的更新信息传递给它的环境变量，因此，在对应的程序文件执行时，可以直接从系统环境变量中获得相关的信息。

具体会传递的信息，包括上下文环境（`Updater.Context`）以及升级信息（`Updater.Context.UpdateInfo`），对应了更新的上下文，和当前的升级信息文件中记录的信息。将会传递的变量信息，只包括对应类型中公开的、类型为 `bool`、`int`、`string` 或 `Version` 的变量。

例如以下脚本：

批处理脚本

```
@ECHO OFF
```

```
CLS
```

```
ECHO 自动更新测试脚本
```

```
ECHO 当前更新的版本是： %CurrentVersion%
```

```
ECHO 更新的原始目录是： %ApplicationDirectory%
```

```
ECHO 更新的地址是： %UpdateDownloadUrl%
```

```
PAUSE
```

在执行的时候会输出这样的结果：



```
C:\Windows\system32\cmd.exe
自动更新测试脚本
当前更新的版本是： 3.0.0.0
更新的原始目录是： C:\Users\iFish\Desktop\Test\tools\..
更新的地址是： H:\Develop\Private\项目发布_\UniversalPackageBuilder3\update_v2\{0}
请按任意键继续。 . . .
```

4.2 注意事项

当填写安装前执行、安装前执行时，请务必注意执行的位置。

当安装程序在执行“安装前执行”时，执行的位置是在新文件的临时目录。当更新程序在进行执行时，更新程序会在系统的临时目录中创建临时的目录，用于保存相关的文件。在安装程序下载升级包文件到临时目录并解压后，等待安装的新文件将位于一个临时的新目录中。安装前执行的程序，将会从这里开始执行。因此，当你在设置增量更新时，请务必将这个进程执行所需要的所有文件，都设置为始终更新，否则会导致执行失败。在执行时若需要访问原程序目录，请使用环境变量来获得相关信息，更多信息参见 [4.1 可用的环境变量](#)。

相反的，“安装后执行”的进程将会从原应用程序目录中启动。

4.3 BUG/建议反馈

本升级库的主页位于：http://www.fishlee.net/soft/simple_autoupdater/

反馈论坛：<http://bbs.fishlee.net/forum-63-1.html>

作者：木鱼 ifish@fishlee.net

微博：<http://t.qq.com/ccfish/> <http://weibo.com/imcfish/>

官方主页：<http://www.fishlee.net/>

QQ 群：[②群 216126338（高级群）](#) [①群 134546850（超级群）](#)