**文档更新履历**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **作成** | **备注** |
| 2019年02月13日 | 1.0 | 曾梓建 | 初版作成 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号： | | | | |
| **标准化名称** | | 基于删除、复制操作的更新程序 | | |
| **适用范围** | | 所有指定路径内的内容（程序、非程序） | | |
| **版本编号** | | **版本履历** | | |
| 1.0 | | 初版作成 | | |
|  | |  | | |
|  | |  | | |
|  | |  | | |
| **使用范例** | | | | |
| 1. 正确放置好主程序与更新程序路径； 2. 编辑UpdataConfig.ini 上的路径； 3. 主程序需完成版本比对，并增加调用、运行更新程序的方法；   详：  [附页1：使用、配置更新程序](#附页1：使用、配置更新程序)  Demo：  [设计网路径](file:///\\192.168.8.104\Share\(20190213)Demo) | | | | |
| **程序流程及说明** | | | | |
| 详：  [附页2：程序运行原理及流程](#附页2：程序运行原理及流程)  该程序配置完成即可使用，无需再次开发。 | | | | |
| **审核时间** |  | | **审核** |  |

附页1：使用、配置更新程序

**前提：**

为便于说明及区分，下文主程序指被更新程序(即需要更新的对象)

**1：程序位置路径**

需要注意:更新程序需要放置在主程序的上一目录

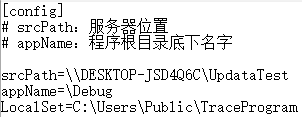


* 图示(Pt\_Program\_Updata.exe):更新程序主体
* 图示(Debug)文件夹内: 主程序的文件夹，被更新的程序路径

**2：配置更新路径**



程序根目录下UpdataConfig.ini 需要在使用前完成配置



* srcPath: 用于存放新版本的路径
* appName: 程序根目录的名字，如上图示(\Debug)
* LocalSet: 程序备份路径

例：

已经编辑过生成好了的程序在：

C:\Users\admin\Desktop\Demo\Application

当前使用中的路径在：

C:\Users\admin\Desktop\(20190213)Demo\Pt\_Program\_Updata\bin\Debug

则：

[config]

srcPath= C:\Users\admin\Desktop\Demo\Application

appName=\Debug

LocalSet=备份路径，可自定义

**3：与主程序绑定**

(当前步骤必需完成1、2步)

在主程序的Load中需要完成版本号的比对，并编写调用更新程序的方法

示例：

|  |
| --- |
| #仅用作示例  string VerResult = "1.0";（程序当前版本）  string Ver\_Upt = "2.4";（程序目标版本）  if (VerResult != Ver\_Upt)  {  DialogResult dr = MessageBox.Show("检测到当前程序有新版本，是否更新？", "提示", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (dr == DialogResult.OK)  {  try  {  //--调用更新程序，自行更新--//  string AppPath = Application.StartupPath + "\\..";  System.Diagnostics.Process exep = new System.Diagnostics.Process();  exep.StartInfo.FileName = AppPath + @"\Pt\_Program\_Updata.exe";  exep.EnableRaisingEvents = true;  exep.Start();  exep.Close();  Environment.Exit(0);  //--调用更新程序，自行更新--//  }  catch (Exception ex)  {  LogWriter lw = new LogWriter();  lw.WriteLog(Convert.ToString(ex));  }  } |

附页2：程序运行原理及流程

**1：程序原理**

主要核心原理如上图，程序主要完成了3个动作，备份→删除→复制。

* 备份：由于考虑到当前程序过程存在不可逆的情况，需要在第一步时执行备份，以避免更新程序执行失败或是中断的情况。
* 删除：当前删除会删除主程序根目录(../XX)下的 **所有** 文件，不含文件夹。
* 复制：从指定路径中复制所有文件到主程序根目录，本质是为了覆盖旧版。

**2 : 程序流程**

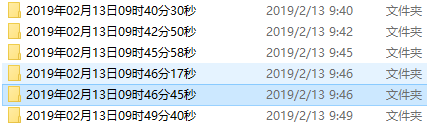
主流程依然如 1:程序原理 中的流程图

**备份：**

1. 文件路径/文件夹命名 定义，声明:

|  |
| --- |
| string Date = DateTime.Now.ToString("yyyy年MM月dd日HH时mm分ss秒");  string LocalSet = con.LocalSet + @"\" + Date; |

备份名称默认示例：



1. 检查路径是否存在

|  |
| --- |
| if (!Directory.Exists(LocalSet))  当结果为true 时，需要程序主动生成一个新的文件夹  Directory.CreateDirectory(LocalSet); |

1. 执行复制任务

|  |
| --- |
| lc.CopyDir(aimPath, LocalSet);  只需要使用年月日，或者其他命名规则做文件夹名时，建议先删除该文件夹，再复制  附件#复制:../LocalControl.CopyDir |

**删除：**

1. 遍历主程序根目录的所有文件

|  |
| --- |
| DirectoryInfo di = new DirectoryInfo(Application.StartupPath + con.appName);  string[] arr = lc.FindFile(di); |

1. 遍历执行删除

|  |
| --- |
| lc.DeleteFile(arr[i]);  附件#删除:../ LocalControl.DeleteFile |

**复制：**

1. 与上文(备份.B)操作相同，注意使用的参数即可

|  |
| --- |
| lc.CopyDir(srcPath, aimPath); |

**注意：**

1. 由于程序存在不可逆的因素，在执行复制动作时，需确保备份动作已完成；
2. 部分电脑进行 新建、删除、复制 的过程中需要获得管理员权限，尽可能避免使用系统盘用作目标、指定路径；必需要使用系统盘时，要将程序加入默认为管理员打开。
3. 若更新程序未做触发条件验证，程序可以单独打开；
4. 在程序运行前，必须要对更新路径做判断，判断路径下是否有程序；