



Visual Studio 扩展插件 用户手册

编写人：李仙伟
2013 年 10 月 26 日



目录

1. 插件简介	1
2. 插件的运行环境	1
3. 插件的安装部署	1
4. 插件使用帮助	2
4.1. 目标平台设置	2
4.2. 输出路径设置	2
4.3. 生成事件设置	3
4.4. NET 版本设置	4
4.5. 折叠和展开所有项目	4
4.6. 批量加载项目	5
4.7. 批量创建项目	5
4.8. 修改项目 dll 引用	6
4.9. 拷贝 dll 的引用 (依赖项)	7
4.10. 更新 dll 到 lib 库	9
4.11. SQL 语句生成器	10
4.12. Guid 生成器	10
4.13. 检查 Lib 库重复的 dll	11



1. 插件简介

基于 Microsoft Visual Studio 2008 环境（具体项目类型为“其他项目类型”→“扩展性”→“Visual Studio 外接程序”），用 C#语言开发实现的 Visual Studio 扩展插件，对 .NET 项目开发很有用，可以批量执行任务，包括折叠和展开所有项目，批量修改项目的目标平台、输出路径、生成事件和 .NET 版本，批量创建和加载项目，快速智能修改项目的 dll 引用、添加 dll 引用、拷贝项目依赖项、查看项目 dll 引用、提交 dll 到 Lib 库、检查 Lib 库是否有重复 dll、生成 SQL 语句和生成 GUID 等，在搭建项目和发布系统时很有用，有了它可以避免很多不必要的问题，同时也可以减少很多重复的工作，尤其是在发布系统时可以保证程序的正确性。在给用户编译发布程序的时候很多人没有意识到目标平台（x86、x64 和 Any CPU）和 .net 版本（2.0 或 3.5 等）的选择对用户可能带来的不良用户体验（通常程序报错，运行不了，或者某个模块运行不了），即便意识到了这一点也很难保证在不同解决方案配置下（debug 和 release 等）程序的目标平台、.Net 版本和 dll 版本的正确性，通常 debug 没有问题，但是一旦发布 release 版本就会出现很多问题，同时如果要手工修改相关配置工作量也很大。该插件绝对是世界上独一无二的，当然如果需要更多批处理功能，聪明的你也可以继续发挥哦……



2. 插件的运行环境

插件基于 Microsoft Visual Studio 2008 环境进行开发，所以插件可以在 Microsoft Visual Studio 2008 以上版本的 Microsoft Visual Studio 环境中使用，目前测试过 Microsoft Visual Studio 2008、Microsoft Visual Studio 2010 和 Microsoft Visual Studio 2012。

3. 插件的安装部署

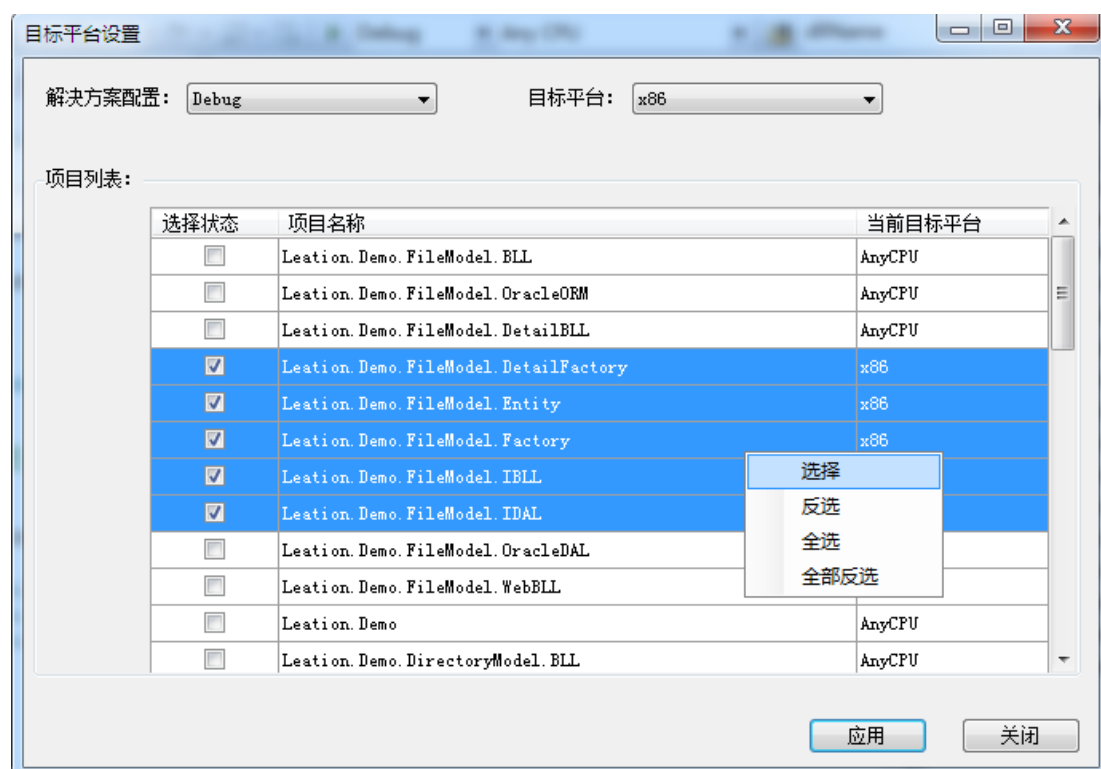


将程序目录“Leation.VSAddin”下的“Leation.VSAddin.dll”和“Leation.VSAddin.AddIn”拷贝到 Visual Studio 外接程序存放目录，例如：“C:\Users\Administrator\Documents\Visual Studio 2008\Addins”、“C:\Users\Administrator\Documents\Visual Studio 2012\Addins”，路径前面部分“C:\Users\Administrator”不同的电脑可能不一样，这是因为您登陆的 windows 账户名不一样，如果不存在“Addins”子目录则需要手动创建该目录，然后启动 Microsoft Visual Studio 即可。

4. 插件使用帮助

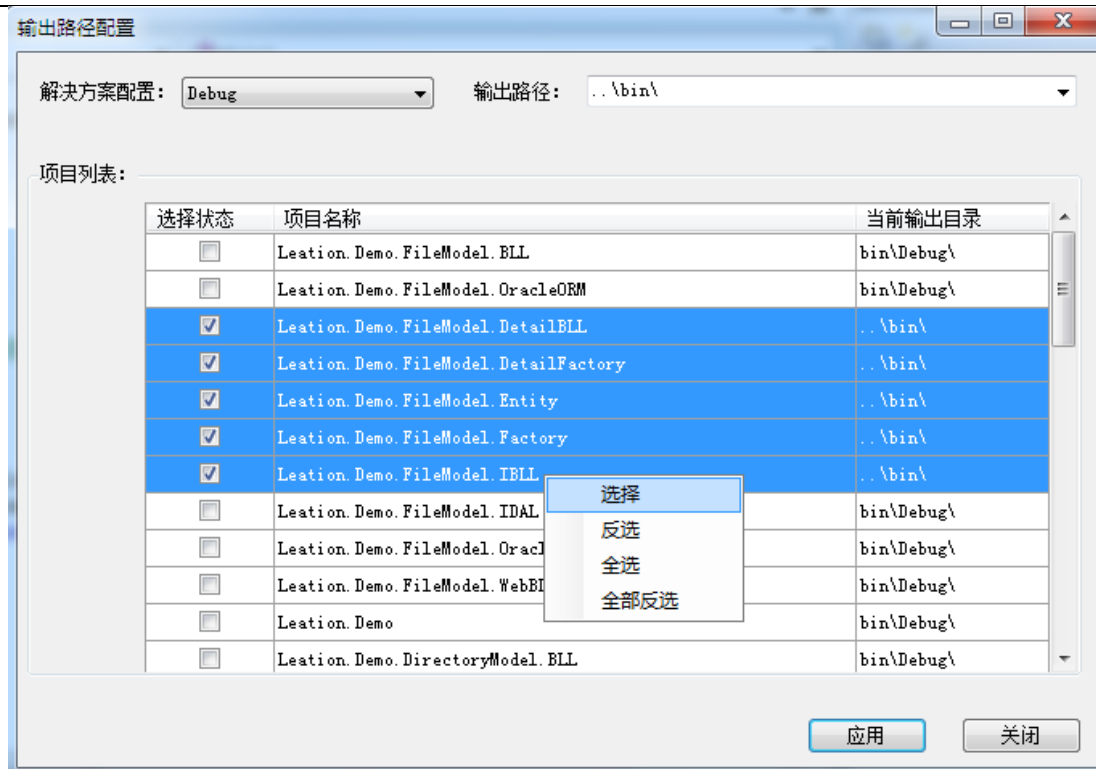
4.1. 目标平台设置

选择一个解决方案配置（一般都是 debug 和 release），选择所需目标平台，勾选需要修改的项目（默认是全部选中），然后点击应用即可修改项目的目标平台，如果修改的项目比较多，可能需要几秒钟甚至更长的时间，**注意：选择在右键菜单里。**



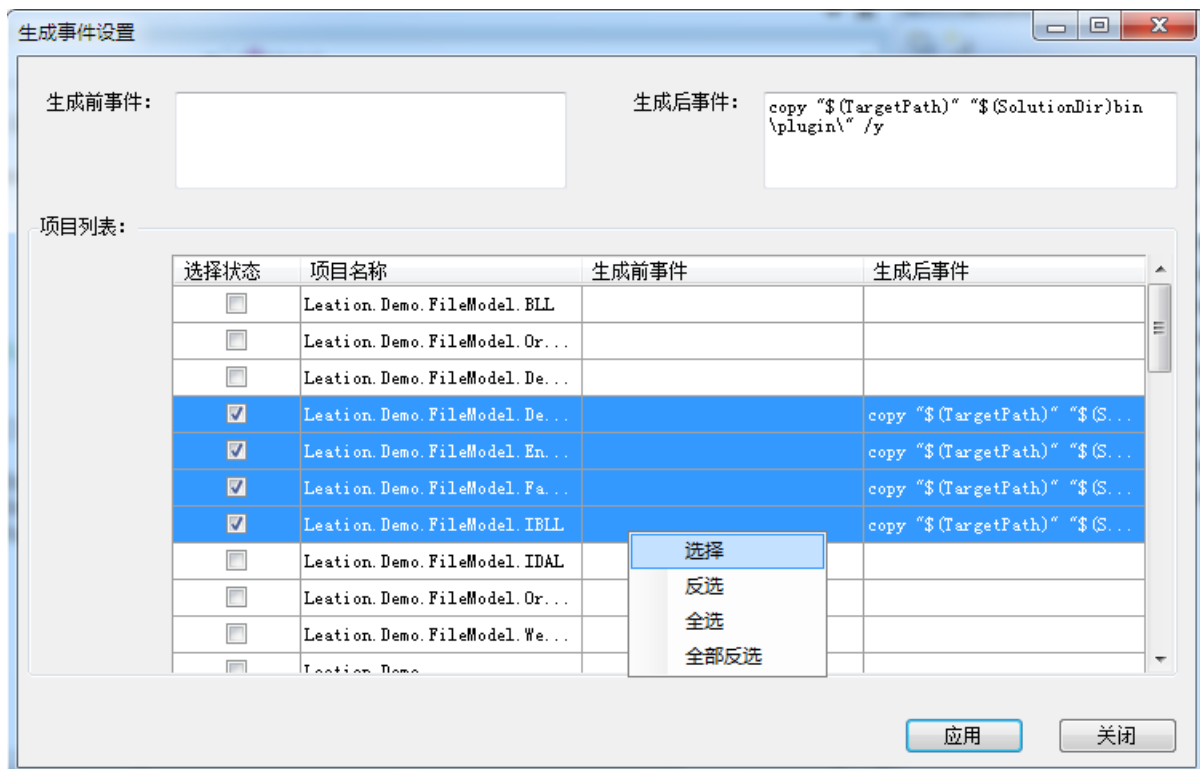
4.2. 输出路径设置

选择一个解决方案配置（一般都是 debug 和 release），输入所需输出路径，勾选需要修改的项目，然后点击应用即可修改项目的目标输出目录，如果修改的项目比较多，可能需要几秒钟甚至更长的时间，**注意：选择在右键菜单里。**



4.3. 生成事件设置

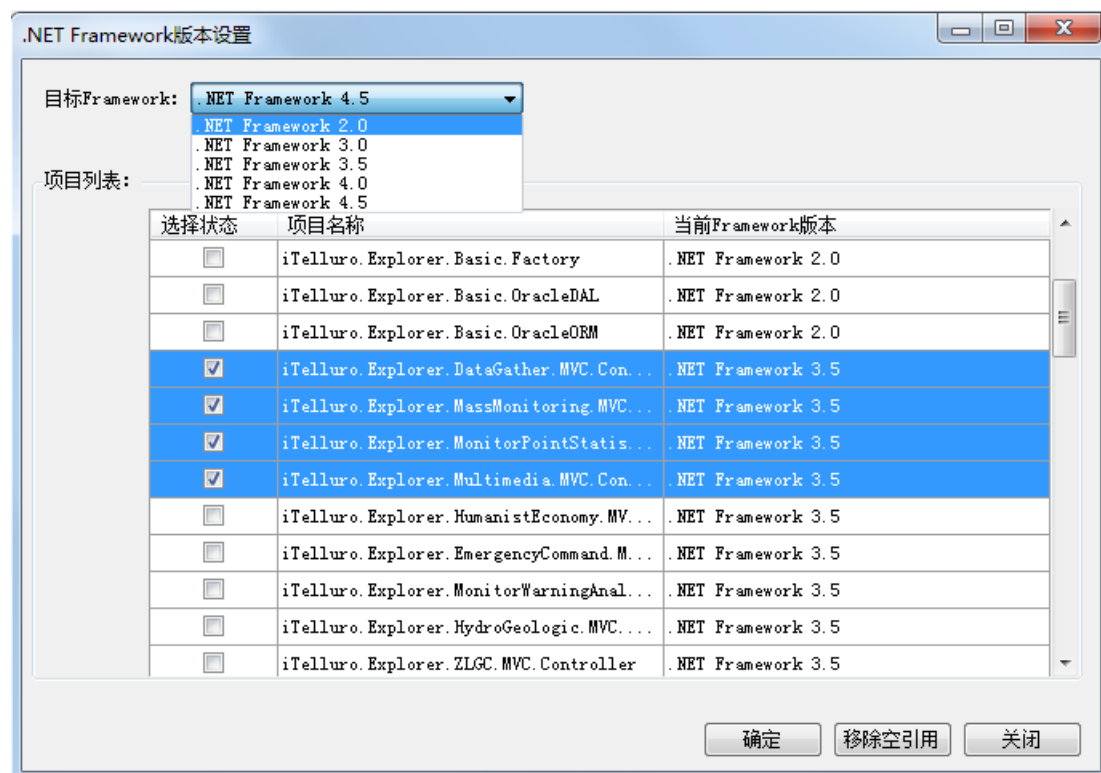
输入所需生成前事件和生成后事件，勾选需要修改的项目，然后点击应用即可修改项目的生成前事件和生成后事件，如果修改的项目比较多，可能需要几秒钟甚至更长的时间，**注意：选择在右键菜单里。**



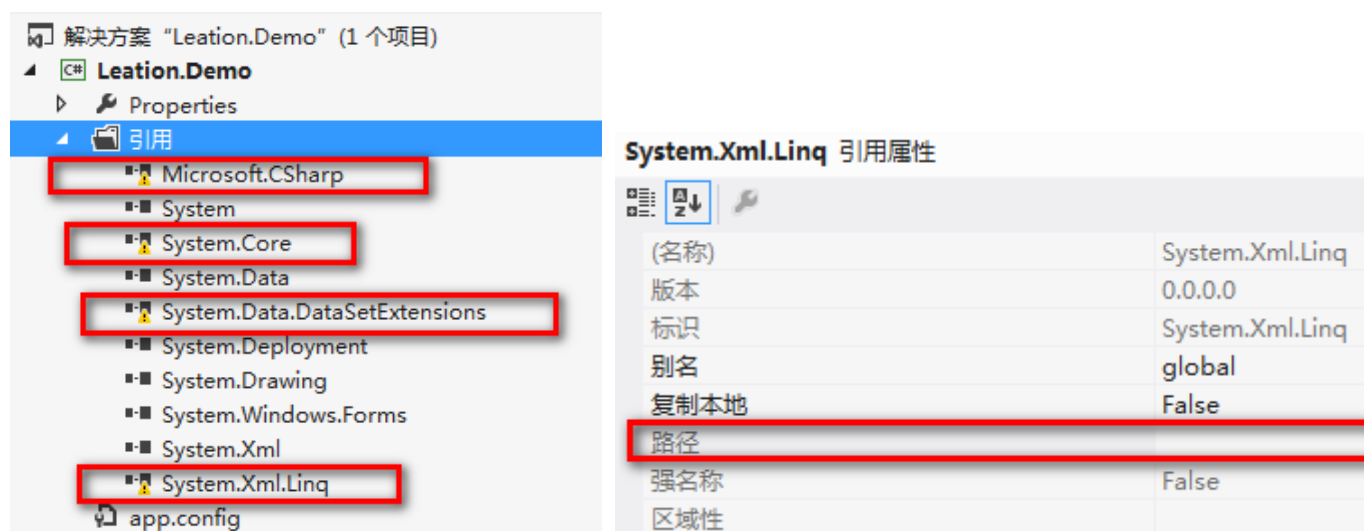


4.4. NET 版本设置

选择目标.NET 版本，勾选需要修改的项目，然后点击“确定”即可修改项目的.NET 版本，如果修改的项目比较多，可能需要几秒钟甚至更长的时间，**注意：选择在右键菜单里。**

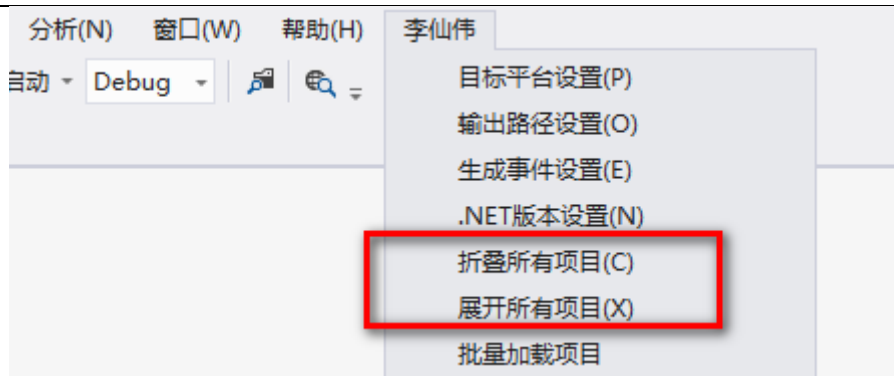


此外，点击“移除空引用”可以移除引用路径为空的应用，这个在将高.NET 版本修改为低.NET 版本时会用到，如下图所示：



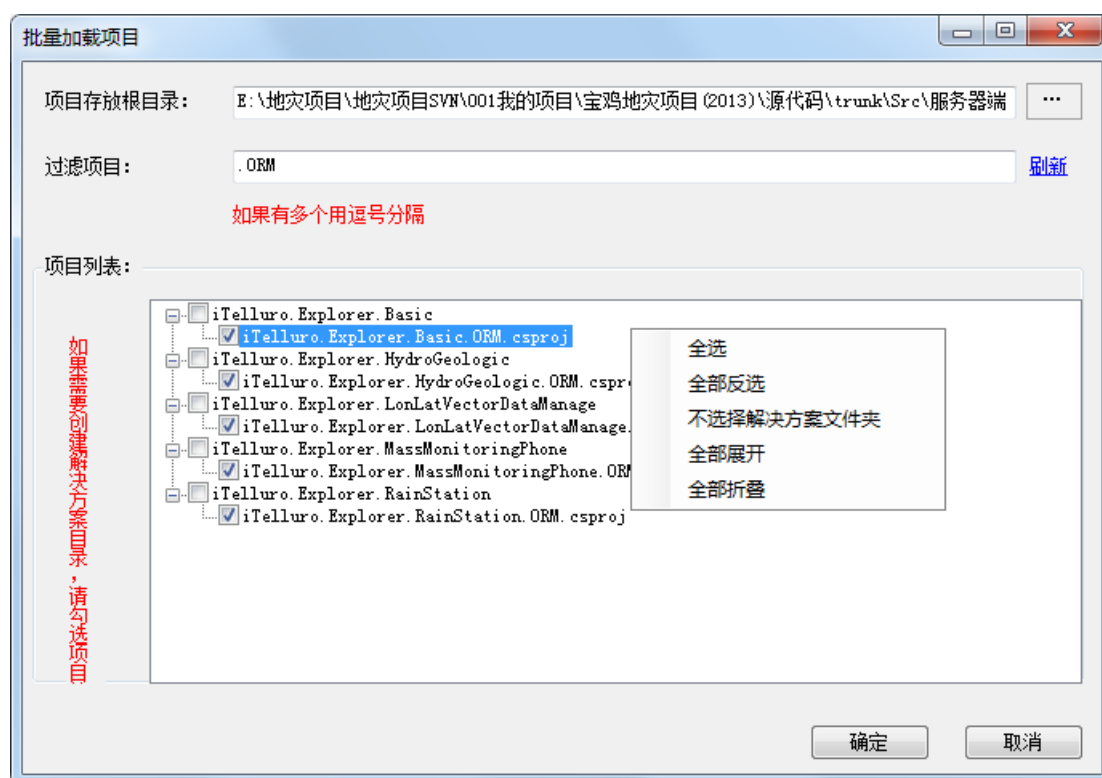
4.5. 折叠和展开所有项目

点击“折叠所有项目”或“展开所有项目”即可折叠或展开当前解决方案下的所有项目和解决方案文件夹。



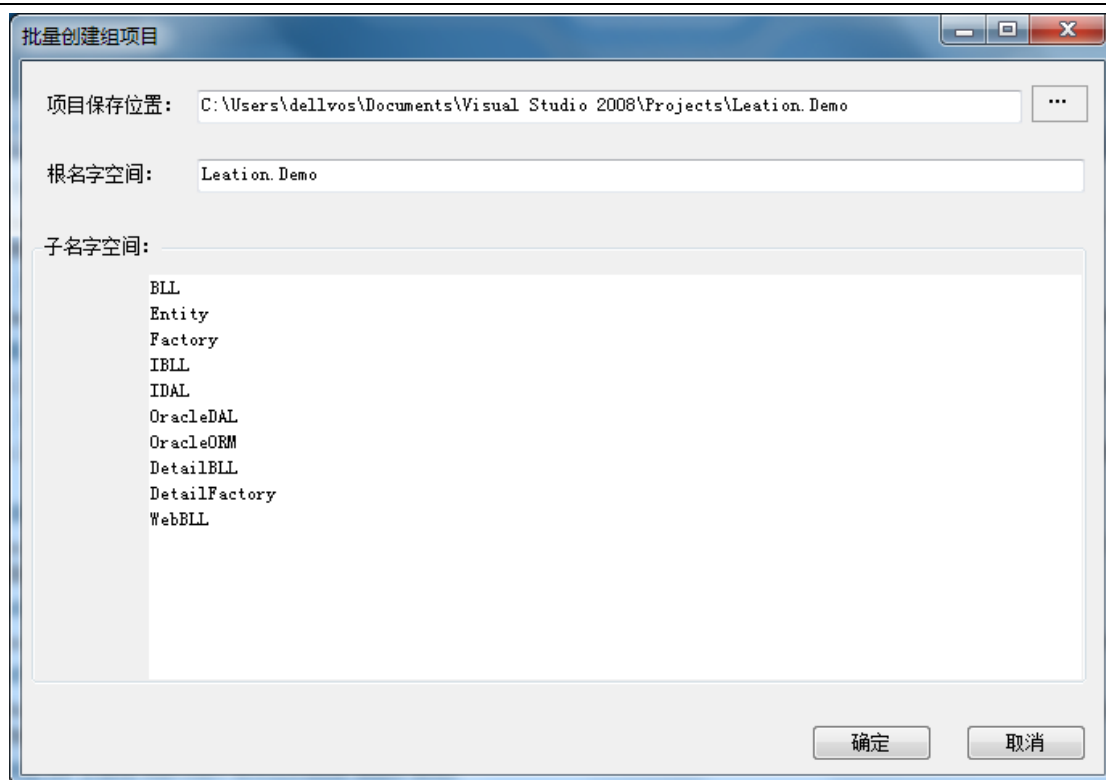
4.6. 批量加载项目

选择项目存放根目录（任意包含项目 (*.csproj) 的目录都可以），然后在项目列表下会列出所有的项目，勾选需要加载的项目，然后点击确定即可加载项目，如果项目太多，可能需要几秒钟甚至更长的时间；如果需要过滤项目，则输入过滤字符串（如 “.ORM”），然后点击“刷新”按钮即可。**注意：选择在右键菜单里。**



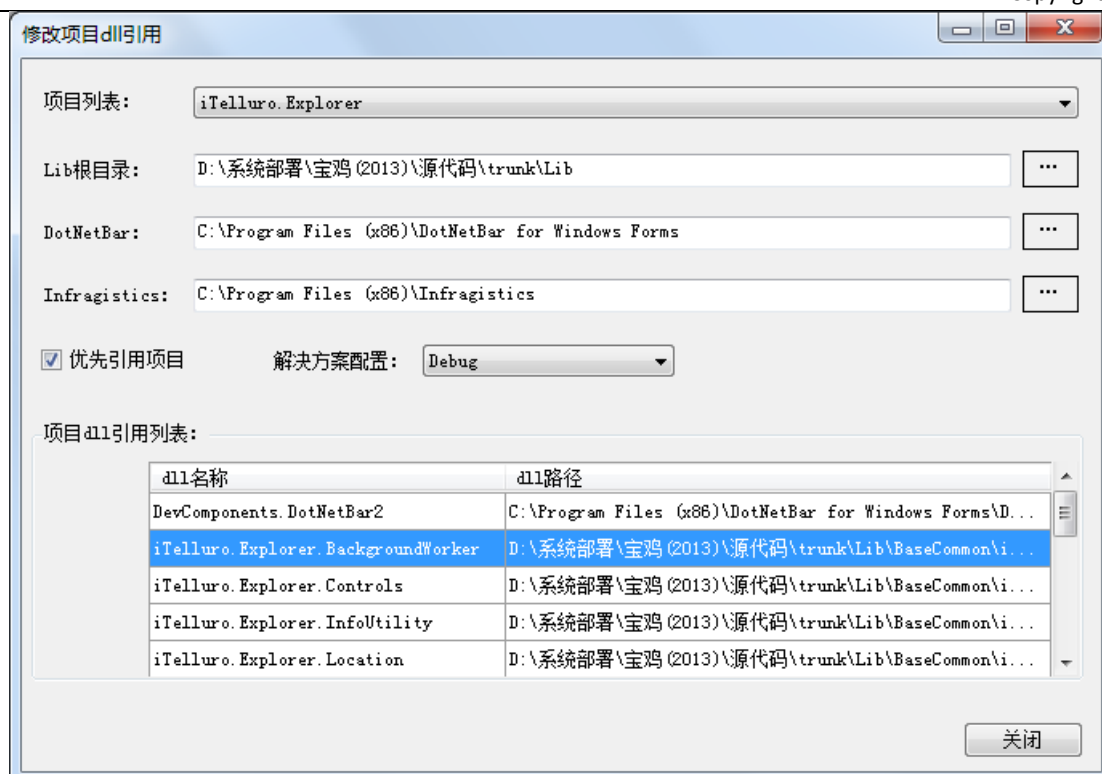
4.7. 批量创建项目

选择项目保存位置，默认是当前解决方案目录，如果需要修改也可以手动修改，一般情况下不需要修改，修改根名字空间，默认是当前解决方案的名称，然后在子名字空间文本框里输入子名字空间，然后点击确定即可批量创建组项目，如果项目太多，可能需要几秒钟甚至更长的时间。

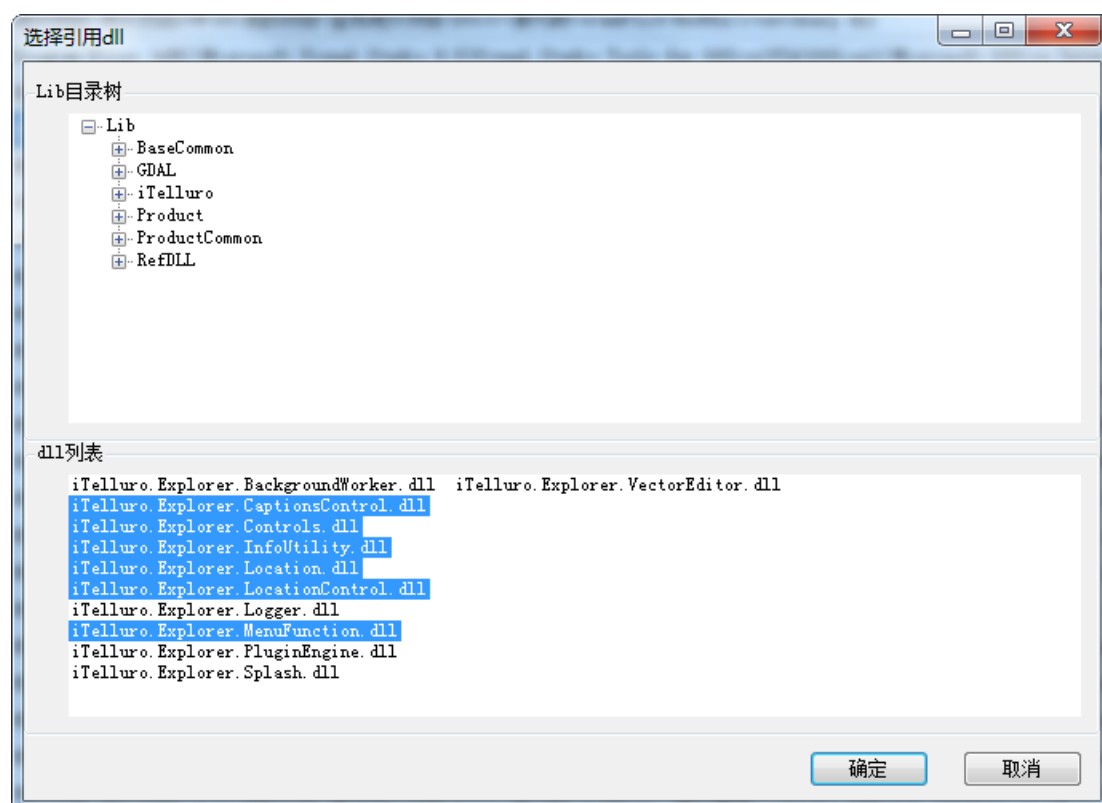


4.8. 修改项目 dll 引用

修改项目的 dll 引用主要是检查项目引用的 dll 是否是指定路径下的 dll 或者同一解决方案下的某一项目，因为当很多人协同开发时，很容易导致 dll 引用路径的错误，虽然程序集名称都一样，但是却是不同的版本，当指定路径下的 dll 更新的时候，项目里引用的其他路径下的 dll，所以有时候各种莫名其妙的问题都会出现。该功能就是为解决这个问题而产生的。如下图所示，选择一个项目，下面表格中会列出该项目引用的 dll 名称及其路径，这样方便检查引用是否正确，同时，可以指定一个 Lib 根目录（即项目里引用的外部 dll 的根目录，程序会自动搜索最佳的 Lib 目录）、DotNetBar 和 Infragistics 三方组件的 dll 路径，然后就可以快速的更新引用 dll 的路径，可以选择某几个 dll，然后进行更新，也可以一键式更新所有项目的 dll 引用路径，如果需要优先引用解决方案下的项目，可以将“优先引用项目”勾上；此外，你还可以添加引用或删除引用。**注意：功能操作在右键菜单里。**



添加引用对话框如下所示，可以很方便快速地添加 Lib 库下所需引用：

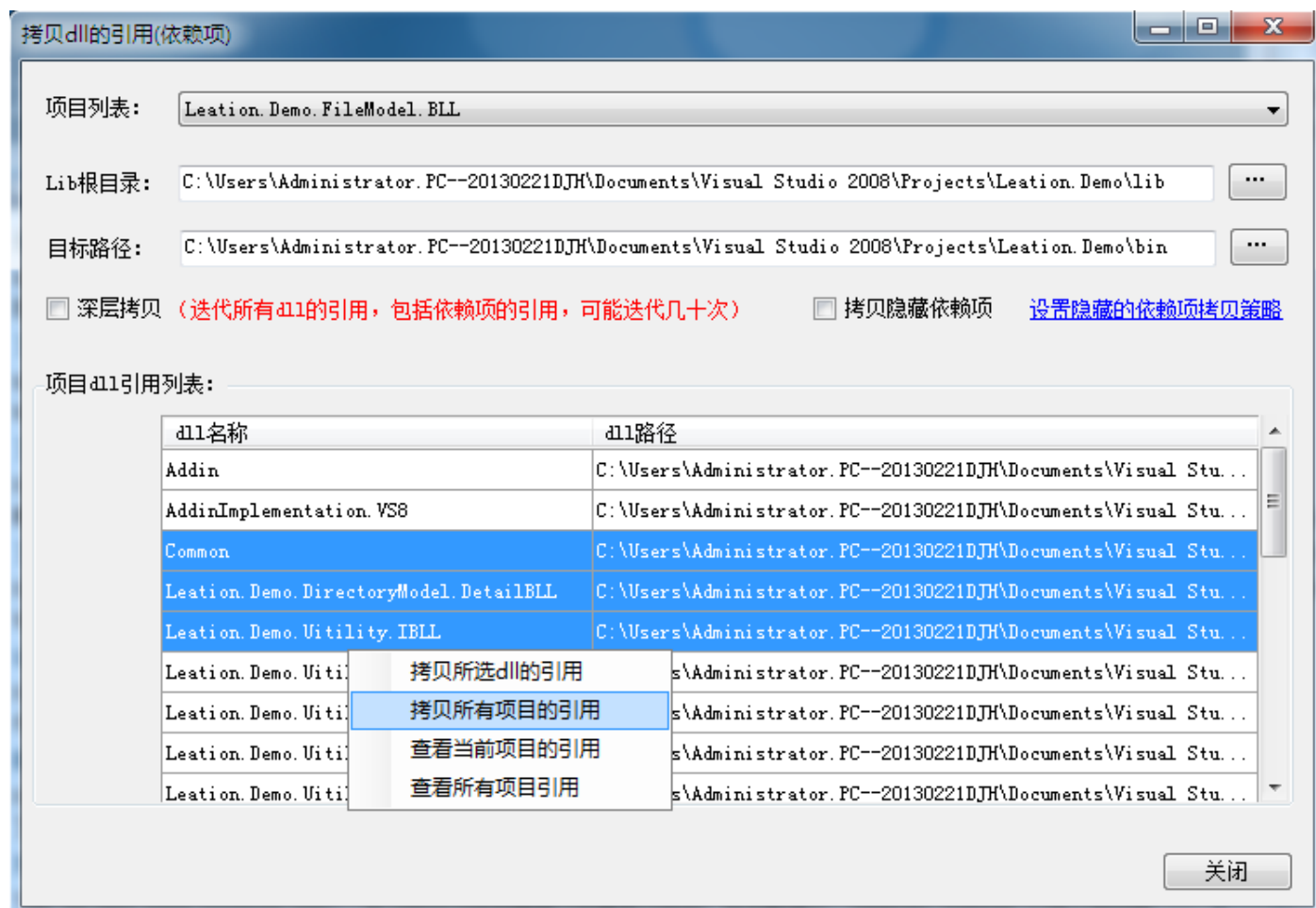


4.9. 拷贝 dll 的引用（依赖项）

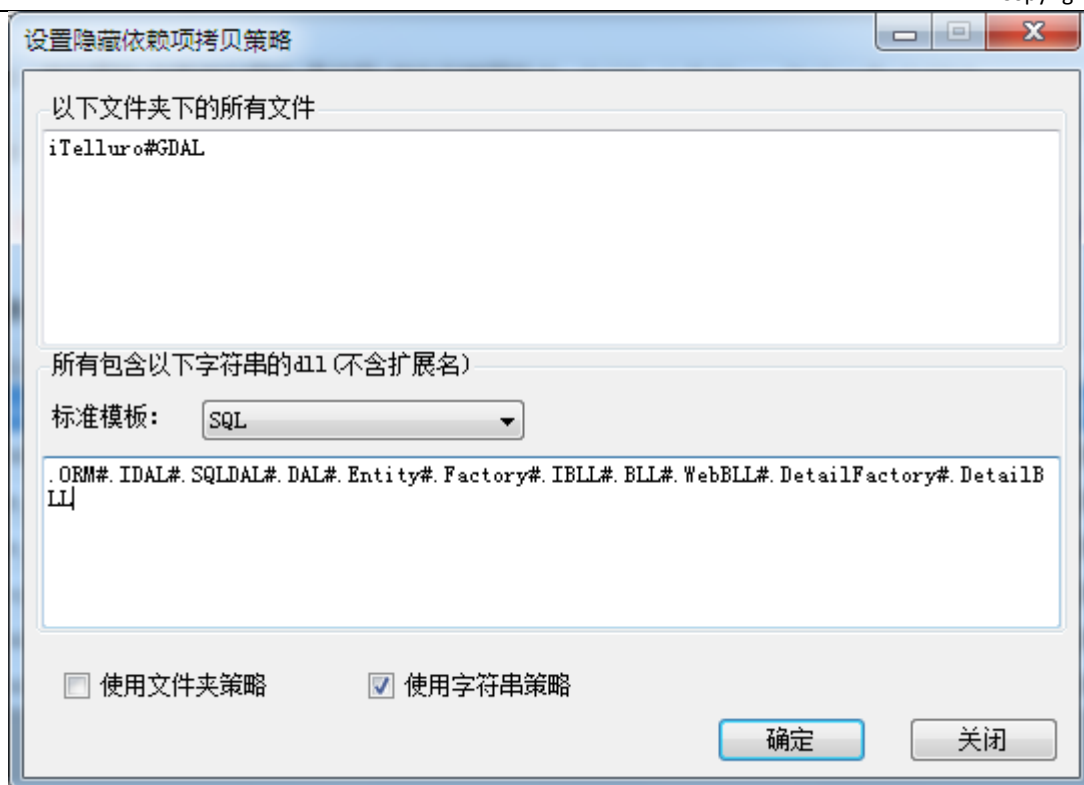
拷贝 dll 的引用（依赖项）主要是快速高效的检查和拷贝项目的依赖项（通常为 dll），包括直接依赖项和间接依赖项，直接依赖项即某个程序集依赖的其他程序集，而间接依赖项则是某个程序集间



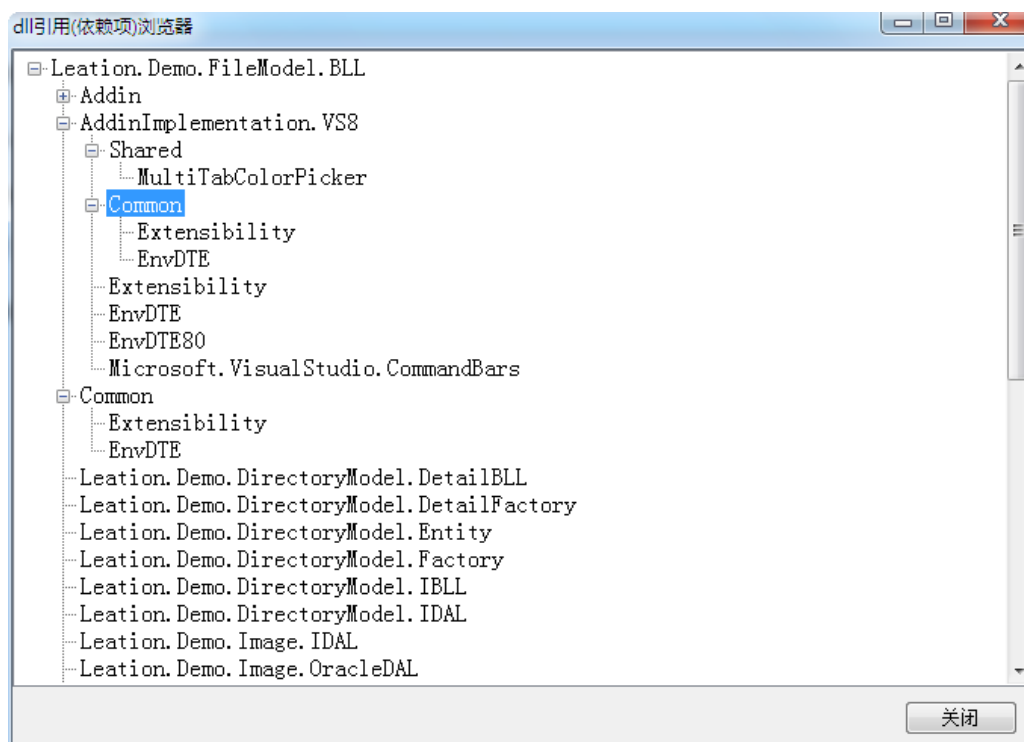
接的依赖项或者程序运行的必须项, 最具代表的是工厂模型, 工厂模型主要是应用了反射原理和接口调用的技术, 一般接口层会单独写到一个程序集里面, 然后接口可以被多个程序集来实现 (比如 SQL 和 Oracle 数据层), 工厂模型通过反射原理进行实例化, 返回一个接口, 具体实例化谁由用户来决定 (比如配置文件), 这样工厂层就不需要引用实现接口的程序集, 所以这就产生了隐藏的依赖项。该工具提供了四个功能项: 拷贝所选 dll 的引用、拷贝所有项目的 dll 引用、查看当前项目的引用和查看所有项目的引用。拷贝 dll 的引用必须先指定 Lib 根目录和目标路径 (程序会自动搜索最匹配的路径), 然后还可以选择是否进行深层拷贝和拷贝隐藏依赖项, 默认为非深层拷贝, 即只拷贝当前项目所引用的 dll 的引用, 注意是引用的引用, 而深层拷贝则会进行迭代, 直到没有引用为止, 当然程序为了避免死循环, 设置了最高 1000 遍历次数。拷贝隐藏依赖项则是通过拷贝策略, 在上述拷贝 dll 的同时检查与此 dll 名称前缀相同, 只是最后一个 “.” 号后面不同的相关 dll, 具体参考策略。



拷贝策略如下所示, 可以设置文件夹策略和字符串策略, 文件夹策略即拷贝指定文件夹下的所有文件 (不含该文件夹下的子文件夹), 字符串策略即在拷贝引用 dll 的同时检查与此 dll 名称前缀相同, 只是最后一个 “.” 号后面不同且最后一个 “.” 后面为所设置的字符串的 dll。注意字符串直接以 # 后分割。



查看 dll 引用如下所示:

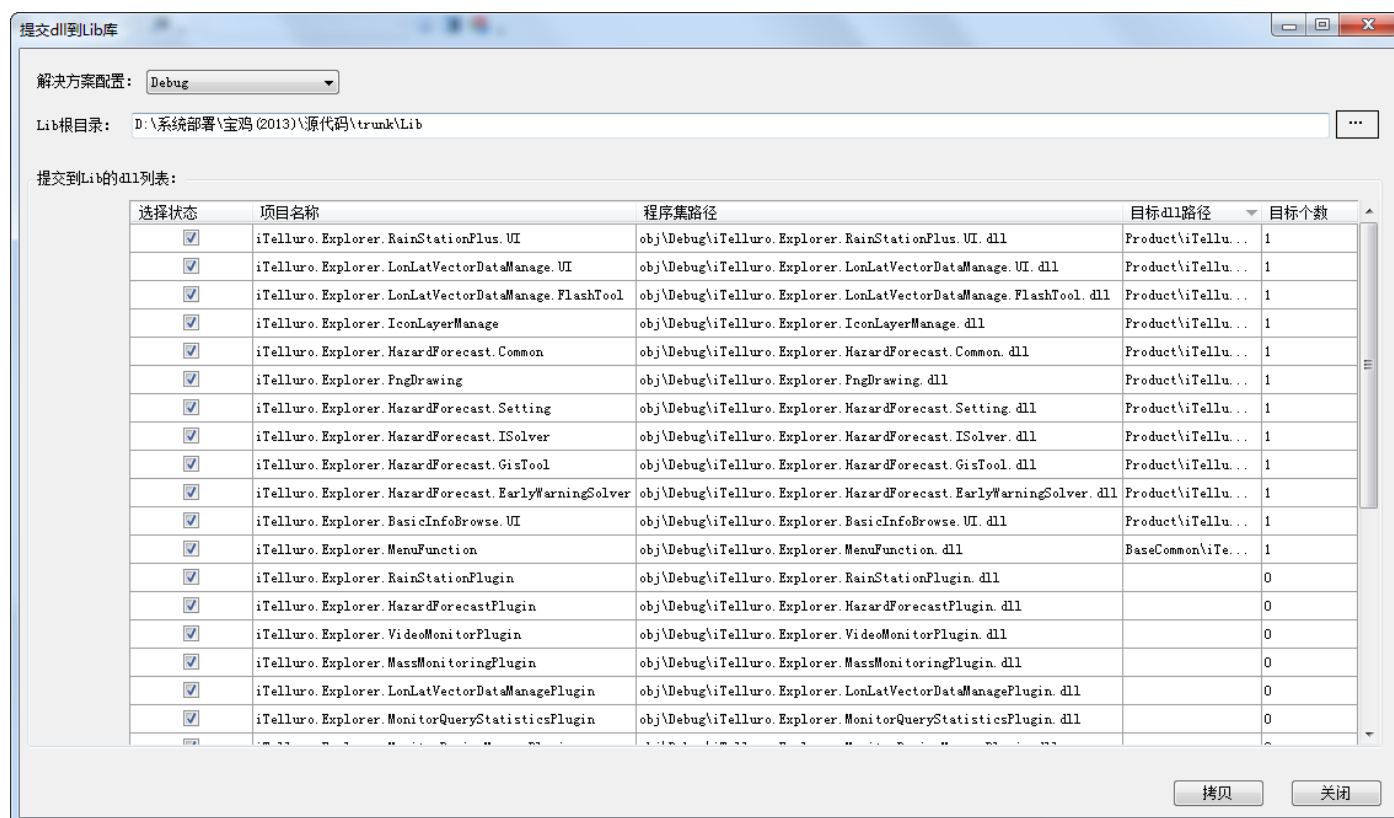


4.10. 更新 dll 到 lib 库

此功能的主要目的是将项目编译生成的 dll 提交到 lib 库下面,以往我们都是手动拷贝,有了这

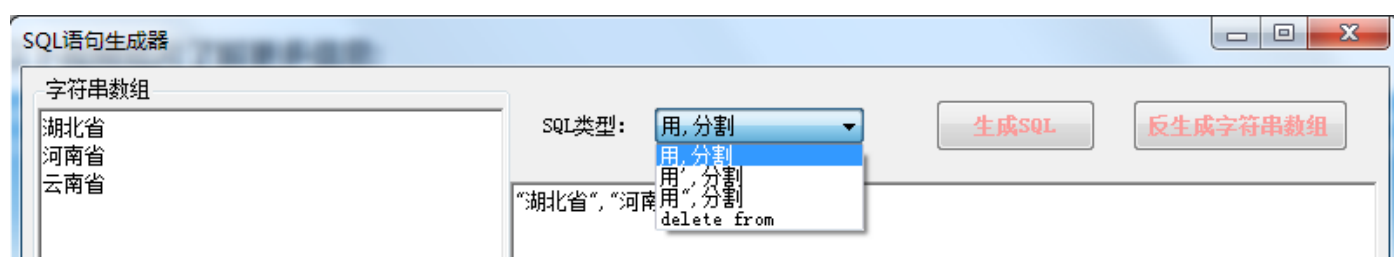


个功能就不需要手动拷贝了，一键式轻松搞定。**注意：选择在右键菜单里。**



4.11. SQL 语句生成器

如下图所示，该功能主要目的是快速的生成写代码或者处理数据时常用到的格式化字符串，比如 SQL 语句，还可以反生产字符串，这是数据处理的利器。

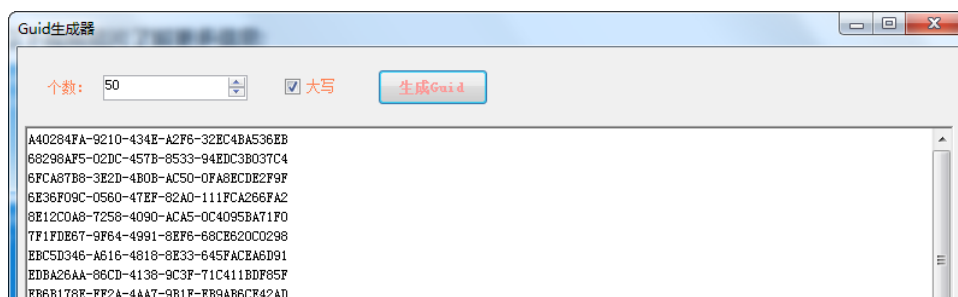


4.12. Guid 生成器

这个工具主要目的就是生成 Guid，VS 已经自带，但是不能批量生成，这里还可以勾选“大写”，这样可以控制生



成的 Guid 字符串的大小写。



4.13. 检查 Lib 库重复的 dll

该功能可以检查 lib 库下有重复的 dll，并可以删除用户选择的 dll。

