**易 包**

1. **什么是易包**

如果说易模块相当于C语言里面的lib文件,则易包就相当于dll文件,也就是说,易模块是在易程序之间建立静态共享,而易包是建立动态共享,避免以前易模块修改后需要重新编译所有使用了它的易程序的问题.

普通的易语言程序使用”编译->编译为易包”菜单功能就可以编译为易包.里面的所有公开子程序将被暴露到其外部供其它程序或易包调用.

所编译的易包是基于机器码的易格式数据,不存在效率问题,也不存在被反编译为易语言源代码的可能.

1. **易包中怎么对外公开子程序**

选中子程序的公开属性即可:



1. **怎么调用其它易包中的公开子程序**

假设我们需要在程序中调用”myepk1”易包内公开的”来自易包1的子程序1”子程序,

需要做以下两件事:

1. 将”myepk1”易包中的该子程序声明部分复制过来;
2. 在其”易包”属性中填入所引用的外部易包名称.

以后就可以像正常子程序一样调用了.

****

请参见”samples\易包\易包及易包之间相互调用”目录内的相关例程

1. **怎么快捷调用其它易包中的公开子程序**

在编译易包时,会同步生成一个同名易模块文件,该模块对外声明了此易包内的所有公开数据(子程序、数据类型、资源、常量、Dll命令等),将此模块导入到需要使用此易包的程序中后,就可以直接访问该易包中的所有公开数据了.

基于此机制,可以避免所发布软件中的易包文件被非法使用(不对外提供对应模块文件即可).

1. **易包的初始化**

易包在被载入的时候会自动载入其启动窗口或调用其启动子程序(根据其所设置的启动方式而定).

具体请参见”samples\易包\易包初始化”目录内的相关例程.

1. **易包与易模块配合使用**

易模块可以考虑将将实际实现代码编译为一个易包,在模块中仅对外提供对此易包的引用,以彻底防止易模块被破解.

具体请参见”samples\易包\易包与易模块配合使用”目录内的相关例程.

另外,编译易包时会同步生成一个易模块文件,将该模块文件引用到你的程序中就可以直接使用易包了.

1. **预载入易包**

默认情况下,在调用外部易包中的子程序时,由系统自动载入相关的易包,但是在某些特殊情况下,可能想覆盖掉此默认操作,在调用外部易包中的子程序前调用相关载入命令即可.

具体请参见”samples\易包\预载入易包”目录内的相关例程.

1. **动态卸载易包**

已显式或隐式载入的易包可以使用”卸载易包”命令进行动态卸载,如果希望所开发的易包支持动态卸载,必须满足以下要求:

任何使用文本/字节集/数组直接常量数据对易包公开子程序以显式或隐式参考方式传递进来的参数进行赋值时,都必须使用"复制易包常量"命令将该数据脱离常量状态,避免本易包被动态卸载后,外部调用程序引用该常量数据出错.

**此规则的另一种阐述方式:**

对于类似 "参数 = 常量" 或 "参数.xx = 常量" 等语句,同时满足以下要求,就需要将"常量"使用"复制易包常量(常量)"代替:

1. "参数"为本易包的公开子程序以显式或隐式参考方式传递进来的参数;

2. "常量"为一个直接的文本/字节集/数组常量

**此规则最简单的记忆方式:**

不修改公开子程序的参数,也不将公开子程序的参数传递到其它子程序,所有需要传递回去的复杂数据均通过一个自定义数据类型返回.

**避免违反此规则的最简单处理方式:**

在公开子程序的首部将所有以显式或隐式参考方式传递过来的参数赋值到一个局部变量中,以后只对这些局部变量进行操作即可.

具体请参见”samples\易包\动态卸载易包”目录内的相关例程.

1. **其它注意点**
2. 静态编译程序中由于核心库代码不全,无法使用易包;
3. 可以直接使用所提供的el.exe(源码已公开)启动易包,也可以使用相关载入命令在易语言程序中启动. 具体请参见”samples\易包\启动易包”目录内的相关例程.