

# 1 系统部署升级要点

## 1.1 系统部署要点

1, 业务逻辑层的程序集文件（比如 XXX.Business.dll）在客户端和服务端上各部署一套，且版本需一致。

2, UI 层的程序集文件（比如 XXX.Window.dll）仅需在客户端上部署，但需与业务逻辑层的程序集版本保持一致。

3, 客户端升级方法，可将客户端用到的程序集拷贝到 Phenix.Services.Host.exe 目录下的子目录 ClientLibrary 里，客户端每次登录时（通过 LogOnDialog）都会自动比对、下载、升级。

以上细节，请参见“使用指南.01.简介”的“系统架构”章节。

## 1.2 系统升级要点

1, 在部署和升级程序集时，需事先通过 Phenix.Services.Host.exe 的“RegisterAssembly”功能菜单，将程序集里的业务类、界面类的元信息注册到 Phenix 配置库中，实现与数据库数据字典的绑定以及初始化权限配置信息。

2, 如果修订了数据库的表结构，比如字符串字段的长度、外键等（即数据库的数据字典），都需通过 Phenix.Services.Host.exe 的“ResetCache”功能菜单，将 Phenix 配置库的元信息与数据库做一次同步。

以上工作务必在系统正式部署和每次升级时都做一遍，否则系统运行当中可能会出现以下故障：

- 1, 在配置系统的功能权限时，找不到业务类、窗体、窗体上的功能组件（比如功能按钮等）等。
- 2, 界面上功能按钮的权限控制不起作用。（参见“使用指南.07.ExecuteAuthorization 组件”）
- 3, 本地业务对象的缓存功能不起作用，比如新增对象且提交成功了，但再次检索后却找不到。

4, 如果删掉了数据库表的物理外键，又未刷新 Phenix 配置库缓存的话，系统可能会正常运行一段时间，直到在无意中刷新缓存，系统会突然抛出“未分析出类{0}与{1}({2})存在关联关系。。。 ”异常提示。（因为 Phenix 并不会实时主动检查数据库表是否已失去了物理外键，开发人员应该在系统升级之前，将子表业务类上的外键字段上打 FieldLinkAttribute 标签，以显式声明存在逻辑外键关系）

5, 如果未注册程序集，或者修改了数据库表字段的长度、不允许为空等条件，又未刷新 Phenix 配置库缓存的话，Phenix 就不会使用到最新的数据库数据字典。这样，业务逻辑层的数据验证、填充默认值等使用到数据字典的功能就不会起作用，会造成比如无法提交数据（比如数据库会抛出某字段不允许为空的异常）等故障。