

新设想VC开发多语言界面支持的新招 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/271/2021_2022__E6_96_B0_E8_AE_BE_E6_83_B3_E2_c97_271878.htm

最近在做一个小程序，要求实现对多语言界面显示支持功能，并且，界面显示内容用户能够自己设置。初步设计用INI文件来配置显示内容，换一种语言的配置文件，就能够更换整个系统的显示语言。考虑到系统规模很小，周期又短，不想用太复杂的方案来解决这个问题，当参考了很多网上类似的设计和代码，发现都不是很满意。主要问题在于：绝大多数基于INI文件配置这种简单应有实现的代码，都是针对组件ID固定加载，写死了组件的ID号，比如：以下是引用片段：
`strCaption = fileManager.GetString(section,"IDC_Stc_ListStudent",""). SetDlgItemText(IDC_Stc_ListStudent,strCaption). strCaption = fileManager.GetString(section,"IDC_Stc_AllContent",""). SetDlgItemText(IDC_Stc_AllContent,strCaption).`这样：界面组件越多，加载代码越长.每新增一个显示窗口，又必须复制、粘贴类似的代码，根据组件ID常量值来修改相关的加载项，很是不爽!初步设想是：设计统一、通用的窗口组件Caption设置方法，对给定的Frame或Dialog等Window容器组件内的所以组件进行遍历，当增、减显示组件不对语言包加载代码产生影响，达到自适应界面组件语言包加载效果。这样就产生一个新问题：语言包配置文件中的Caption值如何跟相关的组件正确地一一对应?好友文国庆建议：用XML文件来定义这种对应关系。这个想法触动了我：反正就是一个[Key,Value]的数据，就用已经实现的INI配置文件也可以啊。于是所有问题

解决! 具体设计是：语言包配置文件就直接设置成组件ID与组件显示信息的Hash表，Key = Value的形式，比如：BtnOK组件的ControlID为“1003”，中文显示Caption为“登录”，语言包配置内容就是“1003=登录”。语言包的加载过程为2步实现：首先，从语言包配置文件中，读取所有配置的ID、Caption条目到Vector或者Array中。其次，在遍历指定窗口中所有组件时，每发现一个组件，就用其ID在已经加载的语言包数组中查找，找到就用配置的值修改组件Caption属性.找不到，就认为是不需要动态配置，不做处理。配置文件实例：
配置项解释：Section：[Login Dialog]：界面窗口.等号左边：窗口中需要设置其Caption属性的组件ID.等号右边：窗口中需要设置其Caption属性的组件Caption值. [Login Dialog] 1001 = 用户帐号 1002 = 用户密码 1017 = 登录 1018 = 退出 语言包配置信息加载代码：以下是引用片段：
BOOL
CLanguageManager::loadFromFile() ...{ BOOL bRead=FALSE. int i. ItemContext temp. CStringArray itemBuf,valueBuf. bRead = fileManager.GetSectionValues("Main Window",itemBuf,valueBuf). if(bRead) ...{ for(i=0;i ...{ temp.uCtrlID = atoi(itemBuf.GetAt(i)). temp.strContext = valueBuf.GetAt(i). m_vtContexts.push_back(temp). } } itemBuf.RemoveAll(). valueBuf.RemoveAll(). bRead = fileManager.GetSectionValues("Login Dialog",itemBuf,valueBuf). if(bRead) ...{ for(i=0;i ...{ temp.uCtrlID = atoi(itemBuf.GetAt(i)). temp.strContext = valueBuf.GetAt(i). m_vtContexts.push_back(temp). } } return bRead. } 100Test 下载
频道开通，各类考试题目直接下载。详细请访问

