

PB 的扩展 DLL 开发（超级篇）（八）

(PB 史上第一次开放的开发技术)

如何调用对象的现有函数和事件

由之前的篇章，我们知道一个 system library 函数，它是 2 个参数，即：(POB_THIS obThis,int nArgs)，如果是这样的一个函数，我们怎么调用它呢？接下来，我举 2 个例子，一个全局函数例子，另一个对象成员函数例子。

我们分析过 OB_THIS 这个结构，PVOID evalued_arglist 成员指向的是一个参数列表，比如 OB_DATA[] 数组，UINT curr_arg_pos 成员指向当前取第几个参数，函数调用时把它初始化为 0，如果有返回值，POT_EVAL_NODE called_return_value 指向一个 OB_DATA，用于接收返回值。OB_INST_ID curr_obinst 设置为当前被调用对象的实例指针。原理就是这么个原理，除此外，还有若干环境需要我们调用前保存，调用后恢复，使用起来相当麻烦。这里，我们知道原理就行了。我们介绍另外的调用方法。要用到 RT_CALL_INFO 结构和 rtRoutineExec、rtReturnValGet 函数。

一、调用全局 MessageBox 函数事例

全局函数，它不依赖于对象实例而存在。

```
int nArgCount = 2; //2 个参数
OB_DATA args[2];
ob_set_data_ptr(&args[0],ob_dup_string(obThis,_T("this is title"))); //标题
ob_set_data_ptr(&args[1],ob_dup_string(obThis,_T("this is message text"))); //显示内容

OB_GROUP_HNDL sysGroupHndl = 0;
OB_CLASS_HNDL hndl = ob_global_reference(obThis, (LPTSTR)_T("MessageBox"), &sysGroupHndl); //获取类句柄
if (sysGroupHndl == 0xffff)
    hndl = ob_global_reference(obThis, (LPTSTR)_T("systemfunctions"), &sysGroupHndl);
//准备调用结构
RT_CALL_INFO ri = { 0 };
ri.bDontTerminateRuntime = TRUE;
ri.enCallType = RT_CLASS_CALL;
ri.rtClassInfo.obClassHndl.class_id = hndl.class_id;
ri.rtClassInfo.obClassHndl.group_hndl = hndl.group_hndl;
//开始调用
RT_EXEC_STATUS status = RT_EXEC_FAILURE;
rtRoutineLevelIncrement(obThis);
OB_VTABLE_ID vTabID = 0;
rtRoutineSearch(m_obThis, ri, (LPTSTR) ("MessageBox"), lpArgs, nArgCount, OB_FUNCTION, &vTabID); //根据参数类型和参
数个数，搜索匹配的函数
status = rtRoutineExec(obThis, ri, lpArgs, nArgCount, vTabID, OB_FUNCTION, TRUE); //调用函数
rtRoutineLevelDecrement(obThis);
POB_DATA returnData = rtReturnValGet(obThis); //取得返回值
至此，全局函数调用完成。
```

二、调用对象实例函数事例，datawindow.show()函数

```
OB_INST_ID obInstID = ....; 这里是 DW 的实例指针

int nArgCount = 0; //0 个参数

OB_DATA args[2];

//如果有参数，请参考全局函数调用方式设置参数

OB_CLASS_HNDL hndl = ob_get_obinst_class_hndl(obThis, obInstDW);

//准备调用结构

RT_CALL_INFO ri = { 0 };

ri.bDontTerminateRuntime = TRUE;

ri.enCallType = RT_INST_CALL;

ri.rtClassInfo.obInst = obInstDW;

ri.rtClassInfo.obClassHndl.class_id = hndl.class_id;

ri.rtClassInfo.obClassHndl.group_hndl = hndl.group_hndl;

//开始调用

RT_EXEC_STATUS status = RT_EXEC_FAILURE;

rtRoutineLevelIncrement(obThis);

OB_VTABLE_ID vTabID = 0;

rtRoutineSearch(m_obThis, ri, (LPTSTR) ("show"), lpArgs, nArgCount, OB_FUNCTION, &vTabID); //根据参数类型和参数个数，搜索匹配的函数

status = rtRoutineExec(obThis, ri, lpArgs, nArgCount, vTabID, OB_FUNCTION, TRUE); //调用函数

rtRoutineLevelDecrement(obThis);

POB_DATA returnData = rtReturnValGet(obThis); //取得返回值

至此，对象实例函数调用完成。
```

如果你对 system library 相关开发方式感兴趣，可到 QQ 群 624409252 共享里大自然的专用目录下下载 DEMO。