跨平台的通用组件库(PhataCommon)

**版本修订批准记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 提交日期 | 版本号 | 版本修订描述 | 作者 | 审核人 |
| 2013-9-17 | V1.0 | 看到淘宝的开源平台（http://code.taobao.org/），发现里面有不少有价值的项目。如ocenabase,tfs,tb-common-utils.  其中tb-common-utils是淘宝一个底层库（其他库会依赖它），提供了线程，线程池，日志，内存，文件，目录，异常，网络通讯等底层的实现。  可惜它只依赖于linux. 平常本人也有收集各种跨平台的通用组件或使用第三方跨平台库，但这些组件或库多多少少都有与当前的需求不太合意的地方。  所以本人决定效仿淘宝自己建一个底层库。因此PhataCommon诞生了！ | phata | phata |
| 2013-9-18 | V1.0 | 添加tree类 | phata | phata |
| 2013-9-25 | v1.0 | 添加atomic | phata | phata |
| 2013-9-25 | v1.0 | 添加mswin\_special目录 | phata | phata |
| 2013-9-25 | v1.0 | 添加可隐藏列的CListCtrl:CListCtrl\_Column\_Picker | phata | phata |
| 2013-10-4 | v1.0 | 删除CListCtrl\_Column\_Picker，添加CGridListCtrlEx(其包含CListCtrl\_Column\_Picker的功能) | phata | phata |
| 2013-10-11 | V1.0 | 添加CButtonST的2个扩展类：CShadeButtonST，CWinXPButtonST。并给它一个单独的文件夹.  添加对话框自动缩放工具EasySize  添加自动缩放对话框CResizableDialog | phata | phata |
| 2013-10-16 | V1.0 | 自己设计CSkinResizeDialog类  添加CXPGroupBox类 | phata | phata |
| 2013-10-23 | V1.0 | 添加CGdipButton类(与CButtonST一样只是复制父窗口的背景，当程序背景一改变或缩放就会有问题)，所以删除CGdipButton类，添加CTraButton类。 | phata | phata |
| 2013-11-08 | V1.0 | 添加线程池spthreadpool | phata | phata |
| 2014-2-25 | V1.0 | 添加进度条类CGradientProgressCtrl | phata | phata |
| 2014-3-1 | V1.0 | 添加数值编辑框CNumericEdit | phata | phata |
| 2014-3-10 | V1.0 | 添加获取CPU信息类CCPU | phata | phata |
| 2014-3-12 | V1.0 | mycrypto改名为phatacrypto | phata | phata |
| 2014-3-21 | V1.0 | 添加authorinfo,vc\_common\_fun | phata | phata |
| 2014-4-2 | V1.0 | 添加base64,url编解码库、CSmtp发送邮件库 | phata | phata |
| 2014-7-18 | V1.0 | 添加FileVersionInfo.h，获取程序版本信息库 | phata | phata |
| 2014-9-22 |  | 添加双滑块控件CRangeSlider |  |  |
| 2014-10-22 |  | 添加NVD库 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

头文件包含如下：

.;$(OPENCV243)/include;$(OPENCV243)/include/opencv;$(GreenSDK);$(GreenSDK)\common;$(GreenSDK)\common\sys

**一、目录结构**

此目录下的代码分为四类：源代码类、本人自己设计的库文件、测试工程与mswin特有的代码。

源代码类内容最多，用处也大。源代码分3种，一种是网上的开源代码(WebOS)，直接可拿来用于工程的；另一种是网上的代码片段，经过本人修改适合本人的使用风格的(WebCode)；最后一种是本人自己写的代码(OwnCode)。

本人自己设计的库文件：平时空余时间捣鼓的库，用于自己的需求，精湛实用。

测试工程：测试本目录下源代码，也为以后使用这些源代码提供例子，测试案例不全，有待持续更新。

mswin特有的代码：前面三类代码都是支持跨平台的，但win平台下有些特性无法跨平台，所以把有价值的，开发中常用到的代码收集起来。

**二、源代码类**

**2.1 WebOS**

**2.1.1 JSON解析库**

使用jsoncpp解析，详情见json/readme.txt。另有rapidjson，google出品。用法见本人博客：<http://blog.csdn.net/wqvbjhc/article/details/9616421>

**2.1.2 log4z日志库**

一直在寻找合适日志库，log4z是目前为止轻型的，支持7\*24记录日志，可按月分目录，每天生成一目录文件。详情见log4z/readme.txt。

**2.1.3 xml解析库**

1) pugixml. 可用于vc6。使用案例为XTE\_IVA。

2) rapidxml. 据说解析速度非常快，使用模板。不用能于vc6。使用案例为XTE\_IVA。用法见<http://blog.csdn.net/wqvbjhc/article/details/7662931>。

**2.1.4 内存池**

tcmalloc内存池，google出品，值得信赖。直接使用源代码比较麻烦，对vs与cb的详细配置见以前项目（XTE\_IVA）.

**2.1.5. 线程管理**

tinythreadpp线程库，实现部分C++11风格，用于不支持C++11的编译器，以便日后代码容易移植。

**2.1.6 tree结构**

STL中没有tree结构，有一项目需要到tree结构。找到了tree.hh: [an STL-like C++ tree class](http://tree.phi-sci.com/)。源码见tree\tree.hh。本人封装了一个操作类，见tree\DeviceTree.h,DeviceTree.cpp。

**2.1.7 CSmtp发送邮件**

主流邮箱商可使用smtp协议发送邮件，但必须使用ssl加密。找到了一个CSmtp类用于发送邮件。

**2.2 WebCode**

**2.2.1 文件目录**

文件操作函数与类，见file/fileunit.h文件。实现简单操作文件功能，有待扩展，未测试。

目录操作函数与类，见file/directoryunit.h文件。实现简单目录操作功能，有待扩展，未测试。

遍历目录的方法，见file/trav\_dir.h,cpp文件。实现遍历目录的几个函数，头文件中有sample.

**2.2.2 INI文件读写类**

见file/INI.h文件。基于模板的。功能较全，未测试。

原地址：<http://code.google.com/p/feather-ini-parser/downloads>

新地址：<https://github.com/Turbine1991/feather-ini-parser>

#include <iostream>

#include "INI.h"

using namespace std;

typedef INI<> ini\_t;

//or

//typedef INI<section\_t, key\_t, value\_t> ini\_t;

ini\_t ini("filename.ini", true);

ini.create("section1"); //Create and select section1

ini.set("key", "value");

cout << ini.get("keynumeric", -1) << endl;

ini["section2"]["key"] = "value";

ini.save();

**2.2.3 guid/xuid生成函数**

见guid/xguid.h文件。使用不同平台的函数实现。

**2.2.4 临界区**

1.有计数的临界区mutex及其辅助类AutoLock. 2.无计数的临界区CriticalSection及其辅助类AutoLockCriticalSection。见sys/mutex.h,mutex.cpp. 使用opencv2.4.6中的Mutex。TinyThread++,log4z也都有自己的互斥体与单例类

**2.2.5 时间函数**

见avaitime.h/avaitime.cpp。实现了getTickCount，getTickFrequency，getCPUTickCount，sleep。

**2.2.6 硬件信息**

见hardwareinfo.h。实现了getNumberOfCPUs。

**2.2.6.1 CPU信息**

win/linux/mac三平台的获取CPU信息类，来自<http://www.cnblogs.com/zyl910/archive/2012/09/29/ccpuid_v102.html>。 有C和CPP两个版本,CPP版本依赖于C版本。源代码为ccpuid.h；ccpuid.hpp，ccpuid.cpp；测试代码为testccpuid\_c.c；testccpuid\_cpp.cpp。

**2.2.7 内存文件映射**

见sharememory.h/sharememory.cpp.未测试。

**2.2.8 自动申请与释放函数类**

见guardclass.h. 可实现定时时自己执行某函数，析构时自动执行某函数。

调用方法

定义类型: typedef int(\*INITFUN)();

typedef CGuardClass<INITFUN,INITFUN> Init\_Clean\_Fun;

构造函数中调用 static Init\_Clean\_Fun ICFun(InitSDK,ClearUp);

一调用就会触发InitSDK函数，静态变量在程序退出时才退出，所以程序退出时才触发ClearUp函数。

**2.2.9** **抽象工厂模式**

见ifactory.h/iporduct.h。只是框架，具体实现见相关项目代码。

**2.2.10 单例模式**

见sys/singleton.h

**2.2.11 线程**

见sys/thread.h。跨平台多线程，被tinythreadcpp取代。

sys/thread\_uv.h,基于libuv实现的多线程函数。

**2.2.12 线程池**

见spthreadpool目录，有C/CPP两个版本。来自<http://code.google.com/p/spserver/>，作者[iunknow](http://iunknown.iteye.com/)。2009年后不见更新，偏向使用C版本。

**2.2.13 原子操作**

见sys/atomic.h.只支持i386和x86,x64架构的cpu. 从淘宝公共代码提取。实现了atomic\_inc/atomic\_add/atomic\_compare\_exchange/atomic\_dec/atomic\_exchange 五种函数。

**2.2.XXX 其他**

如sys/process\_mem\_cpu:进程所占用的内存与cpu.

/// 获取当前进程的cpu使用率(0-100,为百分比)，返回: 0失败，非零成功

int get\_curprocess\_cpu\_usage(double \*cpuusage);

/// 获取当前进程内存和虚拟内存使用量(单位为KB)，返回: 0失败，非零成功

int get\_curprocess\_memory\_usage(uint64\_t\* mem, uint64\_t\* vmem);

MessageBoxTimeout.h--win32 SDK的定时MessageBox

ThreadQueue.h--线程安全的队列

zhao4zhong1\_log.h--CSDN上的日志类。

**2.3 OwnCode**

**2.3.1 编码格式转换**

win平台常用是anscii与unicode相互转换，linux常用是anscii与utf8相互转换。故本人基于C++11实现了：本地编码，utf8编码，unicode编码三种编码格式的相互转换。

添加了base64编解码、url编解码。

**2.3.2 CDialogSK与CResizableDialog的结合类CSkinResizeDialog.h**

把CDialogSK与CResizableDialog所有函数结合在一起。即支持皮肤化，又支持控件自动缩放。源码见mswin\_special\GUI\CSkinResizeDialog.h, CSkinResizeDialog.cpp。

**2.3.3 authorinfo.h、vc\_common\_fun.h、cpp\_common\_fun.h**

autoorinfo.h包含本人的版权信息，希望能编译到二进制文件中，反汇编可看到，但没成功。

vc\_common\_fun.h包含一些VC下的常用功能小函数。

cpp\_common\_fun.h包含一些纯 c/c++下的常用功能小函数。

**三、本人自己设计的库**

**3.1 phatacrypto**

本人设计的密钥管理(基于openssl，但库不依赖它)动态库。详细加解密流程见我的智库文章。

**3.2 videoio**

基于ffmpeg的视频读写库。可读多种视频格式，写只支持avi. lib开头为gcc库。Video开头为vc库。

由它开发出**视频实时流传输工具**。类rtsp流媒体的网络实时视频流库。

**3.3 NVD**

NVD(Network Video Device) 是网络视频设备的简称，该项目是将各种IPC，NVR，DVR，DVS等网络视频采集设备接口统一起来。目前集成了海康与大华2家的接口。相关代码如下：

common/NVD：封装的库，使用**抽象工厂模式**。

common/mswin\_special/NVD：使用MFC封装的播放视频类。

**四、测试工程**

**4.1 Test\_common**

vc与cb测试工程

**五、通用组件库归类**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实现的功能 | 对应的类 | 需要到的文件 | 替代方案 | 备注 |
| 智能指针 | 无 | 无 | stl::shared\_ptr/opencv::Ptr | C++有auto\_ptr,但缺点很多，C++11收入了其他智能指针，shared\_ptr是使用引用计数的。opencv的Ptr也是一个可选择的智能指针 |
| 线程管理 | tthread::thread tthread::mutex tthread::recursive\_mutex tthread::condition\_variable tthread::lock\_guard tthread::fast\_mutex | tinythreadpp/\*.\* | C++11的std::thread | C++11标准有std::thread,但不少编译器不支持，使用TinyThread++库暂时替换 |
| 线程池 | spthreadpool | spthreadpool目录下的相关文件 | ACE/Boost的线程池 | 轻量级，未测试 |
| 内存池 | tcmalloc | tcmalloc/\*.\* | 无 | google的内存池，多项目验证可行 |
| 原子操作 |  | sys/atomic.h | C++11的原子操作 | 淘宝tfs项目中的代码 |
| 日志库 | log4z | log4z/\*.\*(源码包含三个案例) | 无 | 支持按日期分类的日志库 |
| xml解析库 | pugixml.rapidxml | pugixml/\*.\*  rapidxml/\*.\* | 无 | rapidxml快速，优先选择。pugixml可用于不能使用模板的场景 |
| json解析库 | jsoncpp | json/\*.\* | rapidjson号称最快，未测试 |  |
| 互斥体 | Mutex,AutoLock | mutex.\* | 无 | 使用opencv2.4.6中的Mutex。TinyThread++,log4z也都有自己的互斥体与单例类 |
| 单例模式 | Singleton | sys/singleton.h | 无 | 无 |
| 文件操作 | File | file/fileunit.\* | 文件读写或C的 | 无 |
| 目录操作 | Directory | file/directory.\* | 无 | 目录操作功能比较少 |
| 内存映射 | ShareMemory | sharememory.\* | 无 | 无 |
| INI读写类 | INI | file/INI\*.h | 无 | https://code.google.com/p/feather-ini-parser/上的库(1.29版)，wiki有例子 |
| tree类 | tree<> | tree/tree.hh | 无 | 来自<http://tree.phi-sci.com/> |
| excel操作类 | xlslib库，封装了xlslibpacpkage |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**附录一：Windows平台下的库**

追求跨平台是好事，但实际项目中多数还会用到VS IDE,使用MFC做界面。所以对Windows下也有些代码积累，在此列举一下。

**附1.1 界面相关**

**附1.1.1 定时MessageBox**

提示信息一般用MessageBox来显示，但它有一个麻烦就是需要用户去点击才能继续，其实ms提供了一个定时的MessageBox.见mswin\_special\GUI\MessageBoxTimeout.h

**附1.1.2 CDialogSK -- 一个支持皮肤化的对话框类**

可实现MFC对话框的背景贴图，贴图是烦琐，但贴图是最直接的，性能也好。见mswin\_special\GUI\ CDialogSK.h, CDialogSK.cpp.

出处：<http://www.codeproject.com/Articles/4509/CDialogSK-A-Skinnable-Dialog-Class>

本人CSDN介绍：<http://blog.csdn.net/wqvbjhc/article/details/6445356>

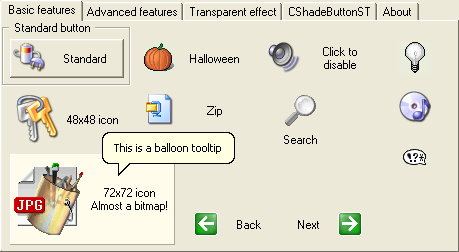
**附1.1.3 CButtonST -- 个性化按钮类**

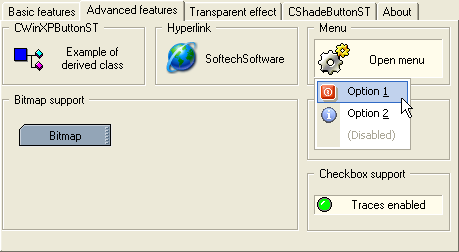
支持贴图的按钮，功能很强大。源码见mswin\_special\GUI\CButtonST目录。

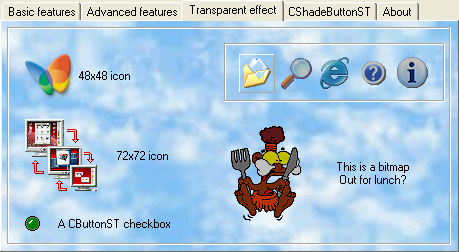
出处：<http://www.codeproject.com/Articles/57/CButtonST-v3-9-MFC-Flat-buttons>

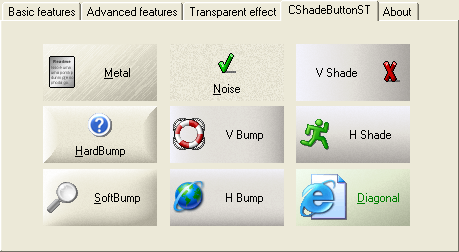
网上他人介绍：<http://blog.csdn.net/liuyang1943/article/details/4959454>

免费版本最高的3.9，之后作者就拿去卖钱了，<http://www.softechsoftware.it/main.html>，现在最新版本4.2.









**附1.1.4 CTraButton—png透明按钮类**

支持透明png贴图的按钮。源码见mswin\_special\GUI\ TraButton.h，TraButton.cpp。CButtonST也有透明功能，但有缺陷：在初始化时使用父窗口设置透明，如果运行时窗口没显示，再显示时透明会出错，不是使用程序的背景去透明，而是使用运行前的窗口背景。

出处：<http://download.csdn.net/detail/duan0783/4964816>有大修改。

支持png,ico,jpg,bmp等图片或资源。但透明只支持png,ico. 有普通，按下，鼠标悬停，不可用四种状态。



**附1.1.5 CGridCtrl -- 一个优秀的网格控件**

类excel风格。本人修改了几处bug. 源码在mswin\_special\GUI\GridCtrl\_src目录和mswin\_special\GUI\NewCellTypes目录。

出处：<http://www.codeproject.com/Articles/8/MFC-Grid-control-2-27>

用法见本人智库与本科毕业设计例子。

**附1.1.6 CGridListCtrlEx - Grid Control Based on CListCtrl**

集合多种功能的CListCtrL控件。与CGridCtrl有些相同的地方。源码在mswin\_special\GUI\ CGridListCtrlEx目录下。

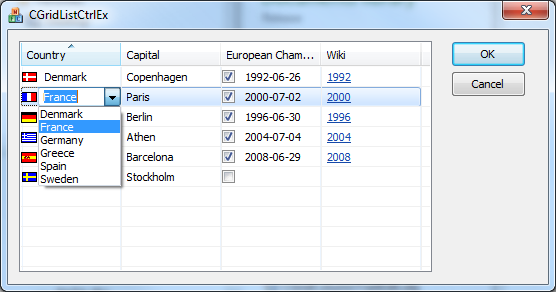
出处：

<http://www.codeproject.com/Articles/29064/CGridListCtrlEx-Grid-Control-Based-on-CListCtrl>

Google code地址：<https://code.google.com/p/cgridlistctrlex/>

Microsoft's **CListCtrl** has support for displaying data in a grid using the report style, but we have to make several changes to implement features like:

* [Sorting](http://www.codeproject.com/KB/list/clistctrl_sort.aspx)
* [Cell navigation and keyboard search](http://www.codeproject.com/KB/list/CListCtrl_subItem_focus.aspx)
* [Tooltips](http://www.codeproject.com/KB/list/CListCtrl_ToolTip.aspx)
* [Hiding and showing columns](http://www.codeproject.com/KB/list/CListCtrl_column_picker.aspx)
* [Cell editing](http://www.codeproject.com/KB/list/nirs2000.aspx)
* [HyperLinks](http://www.codeproject.com/KB/list/hyperlinklistctrl.aspx)
* [Custom row and cell coloring](http://www.codeproject.com/KB/list/lvcustomdraw.aspx)
* [Grouping](http://www.codeproject.com/KB/list/CListCtrl_Grouping.aspx)
* [Clipboard](http://www.codeproject.com/KB/clipboard/clipboard_faq.aspx) (copy only)
* [Persist column width, position and visibility](http://www.codeproject.com/KB/list/clistctrl_persist_state.aspx)
* [OLE Drag and Drop](http://www.codeproject.com/KB/list/explorerdragdrop.aspx) (includes reordering of items)



0.用法

m\_list\_info.*SubclassDlgItem*(IDC\_LIST\_INFO,this);

m\_list\_info.*ModifyStyle*(0,*LVS\_REPORT* | *LVS\_SHOWSELALWAYS* | *LVS\_SINGLESEL*);

m\_list\_info.*SetExtendedStyle*(m\_list\_info.*GetExtendedStyle*() | *LVS\_EX\_FULLROWSELECT* | *LVS\_EX\_GRIDLINES*);

m\_list\_info.SetEmptyMarkupText(*\_T*("无交易记录"));

1. CGridListCtrlEx在headctrl中右键会弹出选择列的菜单（隐藏与显示），这样不利于隐藏列。可重载OnContextMenuHeader，直接返回去掉此功能。

2.设置了LVS\_EX\_FULLROWSELECT属性，选择一列时被选item会focus.其他item会高亮，这样被选item很特殊。可重载SetFocusCell，设置m\_FocusCell = -1;不让任何列focus.

3.设置单元格图标：关联*CImageList*，调用CGridColumnTraitImage::AppendStateImages(\*pList, m\_ImageList);再调用SetCellImage(int nRow, int nCol, int nImageId)

4.列的显示与隐藏ShowColumn(int nCol, bool bShow)

**附1.1.7 XFolderDialog -- CFileDialog实现文件夹的选择**

CFileDialog的扩展类，可实现文件夹的选择。本人修改了几处bug. 源码见mswin\_special\GUI\ XFolderDialog.h, XFolderDialog.cpp.

出处：<http://www.codeproject.com/Articles/9865/XFolderDialog-a-folder-selection-dialog-based-on-C>

**附1.1.8 CImageShowCtrl—显示本地图片的控制，自己负责OnPaint**

CStatic的扩展类，给它一个图片路径，它就能显示出来。内容使用CImage类，支持BMP, GIF, JPEG, PNG, and TIFF几种图片格式。源码见mswin\_special\GUI\ CImageShowCtrl.h, CImageShowCtrl.cpp.

出处：无

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*调用例子\*\*\*\*\*\*\*\*\*\*\*

成员变量定义

CImageShowCtrl m\_showLogoImage;

也控件绑定

DDX\_Control(pDX, IDC\_STATIC\_LOGO, m\_showLogoImage);

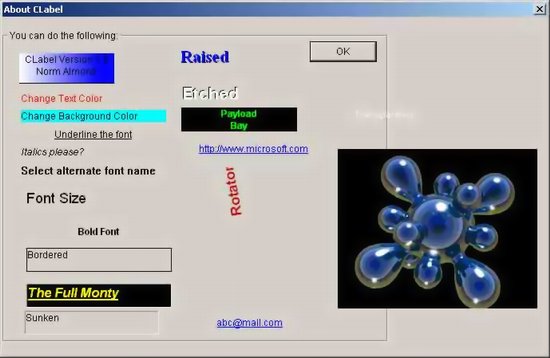
显示：需要时调用下面语句，调用后它会一直显示图片，直到重新调用使用新图片

m\_showLogoImage.ShowPic("1.bmp");

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**附1.1.9 CLabel—功能强大的CStatic类，可设备字体与颜色等**

CStatic的扩展类，可自定义字体颜色等，例子界面如下。



出处：<http://www.codeproject.com/Articles/215/Extended-Use-of-CStatic-Class-CLabel-1-6> （基本版本）

<http://www.codeproject.com/Articles/9064/Yet-Another-Transparent-Static-Control>（参照修改SetWindowsText无法透明的问题）

调用例程

m\_fname.SetFontName("System");

m\_fsize.SetFontSize(14);

m\_uline.SetFontUnderline(TRUE);

m\_tcolor.SetTextColor(RGB(255,0,0));

m\_bcolor.SetBkColor(RGB(0,255,255));

m\_italics.SetFontItalic(TRUE);

m\_bold.SetFontBold(TRUE);

m\_border.SetBorder(TRUE);

m\_sunken.SetSunken(TRUE);

m\_lblGradient.SetBkColor(RGB(255,255,255),RGB(0,0,255), CLabel::Gradient);

m\_monty

.SetFontName("Arial")

.SetFontSize(12)

.SetTextColor(RGB(255,255,0))

.SetFontUnderline(TRUE)

.SetBkColor(RGB(0,0,0))

.SetFontItalic(TRUE)

.SetFontBold(TRUE)

.SetBorder(TRUE)

.SetSunken(TRUE);

m\_flash.SetBkColor(RGB(0,0,0));

m\_flash.SetTextColor(RGB(255,0,0));

m\_flash.SetFontBold(TRUE);

m\_flash.SetTextColor(RGB(0,255,0));

m\_flash.SetWindowText(\_T("Payload\r\nBay"));

m\_flash.FlashText(TRUE);

m\_stcMail.SetMailLink(TRUE,FALSE)

.SetTextColor(RGB(0,0,255))

.SetFontUnderline(TRUE)

.SetLinkCursor(LoadCursor(NULL, IDC\_HAND));

m\_link.SetLink(TRUE,FALSE)

.SetTextColor(RGB(0,0,255))

.SetFontUnderline(TRUE)

.SetLinkCursor(LoadCursor(NULL, IDC\_HAND));

m\_lblTrans.SetTextColor(RGB(255,255,255)).SetTransparent(TRUE);

m\_3D.SetFont3D(TRUE,CLabel::Sunken)

.SetFontName("Arial")

.SetFontSize(16)

.SetFontBold(TRUE);

m\_3D2.SetFont3D(TRUE,CLabel::Raised)

.SetText3DHiliteColor(RGB(0,0,255))

.SetFontName("Times New Roman")

.SetFontSize(16)

.SetFontBold(TRUE);

m\_nAngle = 0;

m\_lblRotate

.SetTextColor(RGB(192,0,0))

.SetFontSize(14)

.SetFontBold(TRUE)

.SetFontName("Arial").SetRotationAngle(m\_nAngle,true);

**附1.1.10 EasySize对话框自动缩放工具**

一个头文件，定义了几个宏，可实现对话框缩放时控件自动缩放. 源码见mswin\_special\GUI\ EasySize.h

出处：<http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time>

1. #include EasySize.h to your stdafx.h (or put it in your include directory and #include <EasySize.h> , which I recommend)
2. Add DECLARE\_EASYSIZE anywhere in your class declaration:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

class CEasySizeDemoDlg : public CDialog

{

DECLARE\_EASYSIZE

...

1. Create an OnInitDialog handler if it doesn't already exist, and put this in the end of it: "INIT\_EASYSIZE;" :

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

BOOL CEasySizeDemoDlg::OnInitDialog()

{

CDialog::OnInitDialog();

...

INIT\_EASYSIZE;

return TRUE; *// return TRUE unless you set the focus to a control*

}

1. Create an OnSize handler and add the UPDATE\_EASYSIZE; macro to it:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

void CEasySizeDemoDlg::OnSize(UINT nType, int cx, int cy)

{

CDialog::OnSize(nType, cx, cy);

UPDATE\_EASYSIZE;

}

1. Optional - If you want your dialog to have a minimum size, then create an OnSizing handler and add theEASYSIZE\_MINSIZE macro as below:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

void CEasySizeDemoDlg::OnSizing(UINT fwSide, LPRECT pRect)

{

CDialog::OnSizing(fwSide, pRect);

EASYSIZE\_MINSIZE(280,250,fwSide,pRect);

}

*//(in this example, 280 is the minimum width and 250 the*

*//minimum height we want our dialog to have)*

1. Now you have to create the "EasySize Map" (or whatever you want to call it) in which you will specify the behavior of each dialog item. It can be placed anywhere inside your class implementation. The map looks like this:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

BEGIN\_EASYSIZE\_MAP(class\_name)

...

EASYSIZE(control,left,top,right,bottom,options)

...

END\_EASYSIZE\_MAP

The map from the demo application looks like this:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/1657/EasySize-Dialog-resizing-in-no-time)

...

*//}}AFX\_MSG\_MAP*

END\_MESSAGE\_MAP()

BEGIN\_EASYSIZE\_MAP(CEasySizeDemoDlg)

EASYSIZE(IDC\_TITLE,ES\_BORDER,ES\_BORDER,

ES\_BORDER,ES\_KEEPSIZE,ES\_HCENTER)

EASYSIZE(IDC\_RADIO1,ES\_BORDER,ES\_BORDER,

ES\_KEEPSIZE,ES\_KEEPSIZE,0)

EASYSIZE(IDC\_RADIO2,ES\_BORDER,ES\_BORDER,

ES\_KEEPSIZE,ES\_KEEPSIZE,0)

EASYSIZE(IDC\_CONTENT,ES\_BORDER,ES\_BORDER,

ES\_BORDER,ES\_BORDER,0)

EASYSIZE(IDC\_STATUSFRAME,ES\_BORDER,ES\_KEEPSIZE,

ES\_BORDER,ES\_BORDER,0)

EASYSIZE(IDC\_STATUS,ES\_BORDER,ES\_KEEPSIZE,

ES\_BORDER,ES\_BORDER,0)

EASYSIZE(IDOK,ES\_KEEPSIZE,ES\_KEEPSIZE,

ES\_BORDER,ES\_BORDER,0)

EASYSIZE(IDCANCEL,ES\_KEEPSIZE,ES\_KEEPSIZE,

ES\_BORDER,ES\_BORDER,0)

EASYSIZE(IDC\_MYICON1,ES\_BORDER,IDC\_RADIO2,IDC\_CONTENT,

IDC\_STATUSFRAME,ES\_HCENTER|ES\_VCENTER)

EASYSIZE(IDC\_MYICON2,ES\_BORDER,ES\_BORDER,IDC\_TITLE,

ES\_KEEPSIZE,ES\_HCENTER)

END\_EASYSIZE\_MAP

///*////////////////////////////////////////////////////////////*

*// CEasySizeDemoDlg message handlers*

...

**附1.1.11 CResizableDialog自动缩放对话框**

可实现对话框缩放时控件自动缩放. 源码见mswin\_special\GUI\ ResizableDialog目录。1.1和1.3版本可用，1.4alpha版本有问题。

出处：<http://www.codeproject.com/Articles/568/CResizableDialog>

Sourceforge地址：<http://sourceforge.net/projects/resizablelib/>（最新版本1.3）

<http://resizablelib.cvs.sourceforge.net/viewvc/resizablelib/?view=tar>(直接下载最新版)

Add the **ResizableLib** to your project's workspace, as explained in the relative [article](http://www.codeproject.com/Articles/1175/ResizableLib).

Create a dialog resource and associate it with a CDialog derived class, for example using Class Wizard, or take a dialog you have already made which you want to be resizable.

You no longer need to change the window style to have the dialog resizing.

Include '**ResizableDialog.h**' in the associated header file.

Search and replace all CDialog occurrences with CResizableDialog in both your .cpp and .h files, just as if your dialog class was derived from CResizableDialog instead of CDialog. I think there's no way to let the Class Wizard do this for you. Let me know if I'm wrong, please.

Your header file should appear like this:

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/568/CResizableDialog)

*// MyDialog.h : header file*

*//*

*// ( Class Wizard stuff )*

///*//////////////////////////////////////////////////////////////////////////*

*// CMyDialog dialog*

#include "ResizableDialog.h"

class CMyDialog : public CResizableDialog

{

*// Construction*

public:

CMyDialog(CWnd\* pParent = NULL); *// standard constructor*

*// ( other stuff )*

*// ...*

}

In your OnInitDialog override, add an anchor for each control you want the size and/or position to be changed when the user resizes the dialog.

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/568/CResizableDialog)

BOOL CMyDialog::OnInitDialog()

{

CResizableDialog::OnInitDialog();

*// Set the icon for this dialog. The framework does this*

*// automatically when the application's main window is not a dialog*

SetIcon(m\_hIcon, TRUE); *// Set big icon*

SetIcon(m\_hIcon, FALSE); *// Set small icon*

*// preset layout*

AddAnchor(IDOK, BOTTOM\_RIGHT);

AddAnchor(IDCANCEL, BOTTOM\_RIGHT);

AddAnchor(IDC\_SPIN1, TOP\_RIGHT);

AddAnchor(IDC\_LABEL1, TOP\_LEFT);

AddAnchor(IDC\_EDIT1, TOP\_LEFT, BOTTOM\_RIGHT);

AddAnchor(IDC\_RADIO1, BOTTOM\_LEFT);

AddAnchor(IDC\_RADIO2, BOTTOM\_LEFT);

AddAnchor(IDC\_GROUP1, BOTTOM\_LEFT, BOTTOM\_RIGHT);

*// other initializations*

*// ...*

Here you may also set a maximum size for your dialog (default is the workspace area), a minimum size (default is the one you set in the resource editor) and also a rectangle that the dialog occupies when maximized.

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/568/CResizableDialog)

*// get desktop size*

CRect rc;

GetDesktopWindow()->GetClientRect(&rc);

*// set max tracking size to half a screen*

SetMaxTrackSize(CSize(rc.Width(), rc.Height()/2));

*// maximized position and size on top of the screen*

rc.bottom = 100;

SetMaximizedRect(rc);

After all this settings, you may wish the dialog's size and position to be automatically saved and restored, as well as its maximized or minimized state. Just provide a Section and an Entry name in your application's profile, in which to save dialog's status.

http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/568/CResizableDialog)

*// save/restore*

*// (for dialog based app, default is a .INI file with*

*// the application's name in the Windows directory)*

EnableSaveRestore(\_T("DemoDlg"));

You are ready to rebuild your project and you will have a resizable dialog just as you wanted.

**1)备注：CResizableDialog子类的注意事项**

我们想要对话框支持控件自动缩放，都是使用CResizableDialog作这父类，但作了子类来说，有几点需要注意的：

**一是**ON\_WM\_SIZE消息中的OnSize函数需要按以下例子添加一些函数才能实现控件缩放。（若没响应ON\_WM\_SIZE消息则不需要）

void Clacesearch\_MFC\_DemoDlg::OnSize(UINT nType, int cx, int cy)

{

CResizableDialog::OnSize(nType, cx, cy);

if (nType == SIZE\_MAXHIDE || nType == SIZE\_MAXSHOW)

return; // arrangement not needed

if (nType == SIZE\_MAXIMIZED)

HideSizeGrip(&m\_dwGripTempState);

else

ShowSizeGrip(&m\_dwGripTempState);

// update grip and layout

UpdateSizeGrip();

ArrangeLayout();

// TODO: 在此处其他代码

}

其中 DWORD m\_dwGripTempState;为私有成员变量，并初始化为1.

**二是**ON\_WM\_ERASEBKGND消息中OnEraseBkgnd函数需要按以下例子添加一些函数才能实现实现对话框背景的显示。（若没响应ON\_WM\_ERASEBKGND消息则不需要）

BOOL Clacesearch\_MFC\_DemoDlg::OnEraseBkgnd(CDC\* pDC)

{

ClipChildren(pDC, FALSE);

BOOL bRet = CResizableDialog::OnEraseBkgnd(pDC);

// TODO: 在此处其他代码

ClipChildren(pDC, TRUE);

return bRet;

}

**附1.1.12 CXPGroupBox—XP样式的GroupBox**

CButton的扩展类，可自定义边框颜色、标题文本颜色、背景颜色、字体等，例子界面如下。



出处：<http://www.codeproject.com/Articles/5573/CXPGroupBox-XP-style-groupbox-and-window-style-gro>

**How to Use it**

I derived the class CXP**GroupBox** from CButton. It is easy to use. In the dialog resource editor, add a **groupbox** as usual and give an ID to the **groupbox**. Then use class wizard to create a member to link the **groupbox**. Remember, select category as "Control", and variable type as CButton.

Then go to Dialog header file and add:

#include "XP**GroupBox**.h"

and also change the control type CButton to CXP**GroupBox**. For example, change:

CButton m\_frmXP**GroupBox**;

to

CXP**GroupBox** m\_frmXP**GroupBox**;

Then in the CPP file of the dialog class, in the OnInitDialog() function, add init code for the **groupbox**. In the demo file, it shows:

m\_frmWin.SetXPGroupStyle(CXP**GroupBox**::XPGB\_WINDOW)

.SetBackgroundColor(RGB(201, 201, 151), RGB(255, 255, 204))

.SetFontBold(TRUE);

m\_frmNormal.SetCatptionTextColor(RGB(16,140,231))

.SetFontBold(TRUE);

m\_frmWin is the new style **groupbox**, it looks like a window and m\_frmNormal is XP style **groupBox**.

Enjoy it.

**Class function list**

virtual CXP**GroupBox**& SetBorderColor(COLORREF clrBorder);//边框线的颜色

virtual CXP**GroupBox**& SetCatptionTextColor(COLORREF clrText);

virtual CXP**GroupBox**& SetBackgroundColor(COLORREF clrBKClient);

virtual CXP**GroupBox**& SetBackgroundColor(COLORREF clrBKTilte,

COLORREF clrBKClient);

virtual CXP**GroupBox**& SetXPGroupStyle(XP**GroupBox**Style eStyle);

virtual CXP**GroupBox**& SetText(LPCTSTR lpszTitle);

virtual CXP**GroupBox**& SetFontBold(BOOL bBold);

virtual CXP**GroupBox**& SetFontName(const CString& strFont,

BYTE byCharSet = ANSI\_CHARSET);

virtual CXP**GroupBox**& SetFontUnderline(BOOL bSet);

virtual CXP**GroupBox**& SetFontItalic(BOOL bSet);

virtual CXP**GroupBox**& SetFontSize(int nSize);

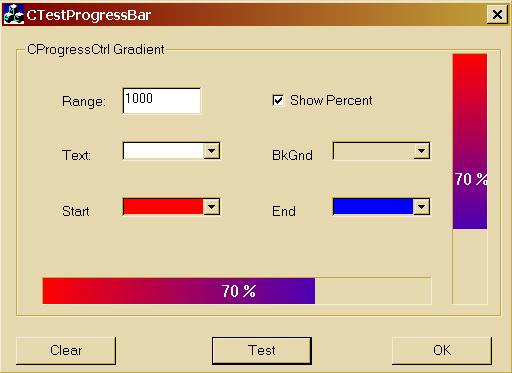
virtual CXP**GroupBox**& SetFont(LOGFONT lf);

In the class, you can set font attributes like font name, color, size, bold, italic etc. You can change the background color of the window style **groupbox** and XP style **groupbox** as well. If you change the background color of**groupbox**, you will have to make the style of the controls in the **groupbox** transparent. Otherwise it is urgly. :)

备注：放在GroupBox上的控件有时会被遮住看不到，解决办法为用记事本打开.rc文件，把GROUPBOX的定义放在所有控件最前面。

**附1.1.13 CGradientProgressCtrl—丰富的进度条**

CProgressCtrl的扩展类，颜色渐变，可自定义起始颜色，背景颜色，字体颜色等，例子界面如下。



**附1.1.14 CNumericEdit—只允许输入数值的编辑框**

//使用方法

//在你的对话框中添加变量：

//private:

//CNumericEdit m\_Edit;

//在对话框的OnInitDialog()初始化！

// BOOL CAboutDlg::OnInitDialog()

//{

// ...

// //控件的子类型化

// m\_Edit.SubclassDlgItem(IDC\_YOUR\_EDIT,this);

// ....

//}

//--------------

// 别忘了包含相应的头文件！

**附1.1.15 CEnhanceEdit—增强型编辑框，可设字体颜色，提示，只接受数值等**

//使用方法

//在你的对话框中添加变量：

//private:

// CEnhanceEdit m\_Edit;

//在对话框的OnInitDialog()初始化！

// BOOL CAboutDlg::OnInitDialog()

//{

// ...

// //控件的子类型化

// m\_Edit.SubclassDlgItem(IDC\_YOUR\_EDIT,this);

// ....

//}

//--------------

// 别忘了包含相应的头文件！

**附1.1.16 CFileVersionInfo—获取文件版本信息类**

出处：<http://www.codeproject.com/Articles/1502/CFileVersionInfo-Retrieving-a-File-s-full-Version>

**附1.1.17 CRangeSlider—双滑块控件**

出处关喜记质量诊断客户端。~~需要用到MeMDC,与其他项目的memdc冲突，未设置好。~~把memdc写在头文件中，并加入namespace,解决memdc冲突问题。

用法如下：

m\_brightSlider.LoadChannelImage(*RT\_BITMAP*, IDB\_BITMAP\_NORMAL, IDB\_BITMAP\_ABNORMAL);

m\_brightSlider.LoadThumbImage(*RT\_BITMAP*, IDB\_BITMAP\_WARN);

m\_brightSlider.SetThumbMargin(12);

m\_brightSlider.SetChannelMin(0);

m\_brightSlider.SetChannelMax(255);

AnalyzeRule rule;

rule.eAlgorithmType = algorithm\_bright;

rule.fMinValue = m\_brightSlider.GetChannelMin();

rule.fMaxValue = m\_brightSlider.GetChannelMax();

rule.fWarningThreshold = m\_brightSlider.GetThumbMin();

rule.fAbnormalThreshold = m\_brightSlider.GetThumbMax();

**附1.2 系统相关**

**附1.2.1 win服务操作类**

来自安络大成的视频运维项目中mysql服务重启类，未修改成通用的。 源码在mswin\_special\sys\WinServices.h,WinServices.cpp。

**附1.3 其他**

**附1.3.1 操作Excel**

见mswin\_special\excel9.h,excel9.cpp,CExlRep.h,CExlRep.cpp。--被xlslib取代