# 这是c++.net的总结

# checkbox绑定通过ds读取xml

String^ dir=String::Format("{0}\\config\\aaa.xml",Directory::GetCurrentDirectory());

DataSet^ ds1=gcnew DataSet;

ds1->ReadXml(dir);

for each(DataTable^ dt in ds1->Tables)

{

if(dt->TableName!="b")

{

continue;

}

for each(DataRow^ dr in dt->Rows)

{

String^ dir = String::Format("{1}\\{0}", dr["bb"],

Directory::GetCurrentDirectory());

if (!Directory::Exists(dir)) {

continue;

}

cbo\_bing->Items->Add(dir);

}

第一：获取节点

【

如果xml中内容是：<?xml version="1.0" encoding="gb2312"?>

<root>

<b><bb><a1> a </a1></bb></b>

<b><bb><a2>a22</a2></bb></b>

</root>

则 ds1->Tables有两个即 b和bb 。 如果

xml内容是：

<root>

<b><bb1><a1> a </a1></bb1></b>

<b><bb><a2>a22</a2></bb></b>

</root>

则 ds1->Tables有三个即b，bb，bb1。

如果xml内容是：

<root>

<b><bb1>a </bb1></b>

<b><bb2>a22</bb2></b>

</root>

则 ds1->Tables有一个即b。

《长城总结：应该是如果加上root有大于两个节点的，那么ds1->Tables的名字应该是从root一下到最后一个节点以上都是。

如果加上root等于两个节点的，则ds1->Tables的名字只有root一个。我测试是这样的。》】

第二：获取节点值：

xml内容如下：

<?xml version="1.0" encoding="gb2312"?>

<root>

<b><bb> a</bb></b>

<b><bb>a22</bb></b>

</root>

后台代码绑定checkbox如下：

mFlash->loadBin(fileName);

String^ dir=String::Format("{0}\\config\\aaa.xml",Directory::GetCurrentDirectory());

DataSet^ ds1=gcnew DataSet;

ds1->ReadXml(dir);

for each(DataTable^ dt in ds1->Tables)

{

if(dt->TableName!="b")

{

continue;

}

for each(DataRow^ dr in dt->Rows)

{

cbo\_bing->Items->Add(dr["bb"]);

}

}

# checkbox绑定通过枚举类型

在.h文件添加

};

public enum class Partition {

boot,

dsp,

app,

fs

};

}

using namespace secApp;

注意是在最后一个括号前面添加。

在.cpp调用是：

comboBox1->DataSource = Enum::GetValues(Partition::typeid);

【注意：Paritition可以定义在类库中，这样也能获取到。】

# L和\_T()

这个是从网上搜集的材料：

L "www.microsoft.com ", 那个L是什么意思

L "www.microsoft.com "

L的意思就是说后面的字符串为宽字符文字，为了支持汉语等语言

L是将后面的字符或字符串转换成相应的 Unicode 形式

L指的是UNICODE字符集的宽字符,每两个字节表示一个字符,用来支持世界所有的语言,原来的ANSI字符集是单字节集,不能很好的支持世界上的两字节语言集,所以现在编写国际化的程序软件时都需要使用UNICODE字符集.

C++中L和\_T()之区别 .

分类： C++ 2011-08-17 15:30 180人阅读 评论(0) 收藏 举报

字符串前面加L表示该字符串是Unicode字符串。

\_T是一个宏，如果项目使用了Unicode字符集（定义了UNICODE宏），则自动在字符串前面加上L，否则字符串不变。因此，VisualC++里边定义字符串的时候，用\_T来保证兼容性。VC支持ascii和unicode两种字符类型，用\_T可以保证从ascii编码类型转换到unicode编码类型的时候，程序不需要修改。

以下是别人的总结：

一、在字符串前加一个L作用:

如 L"我的字符串"表示将ANSI字符串转换成unicode的字符串，就是每个字符占用两个字节。

strlen("asd") = 3;

strlen(L"asd") =6;

二、\_T宏可以把一个引号引起来的字符串，根据你的环境设置，使得编译器会根据编译目标环境选择合适的（Unicode还是ANSI）字符处理方式

如果你定义了UNICODE，那么\_T宏会把字符串前面加一个L。这时 \_T("ABCD") 相当于L"ABCD" ，这是宽字符串。

如果没有定义，那么\_T宏不会在字符串前面加那个L，\_T("ABCD") 就等价于"ABCD"

三、TEXT,\_TEXT 和\_T 一样的

如下面三语句：

TCHAR szStr1[] = TEXT("str1");

char szStr2[] = "str2";

WCHAR szStr3[] = L("str3");

那么第一句话在定义了UNICODE时会解释为第三句话，没有定义时就等于第二句话。

但二句话无论是否定义了UNICODE都是生成一个ANSI字符串，而第三句话总是生成UNICODE字符串。

为了程序的可移植性，建议都用第一种表示方法。但在某些情况下，某个字符必须为ANSI或UNICODE，那就用后两种方法。

别人的总结2：

你要确定你需要的字符串是宽字符还是窄字符。\_T("")是说如果你定义了UNICODE那么就是L"",没有定义就是"";以下情况用\_T（）比较好，其他情况最好别用：

1，用THCAR，LPTSTR，LPCTSTR等tchar数据类型的时候

2，用\_tprintf之类的\_t版本运行时函数时候

3，像下面这样有w和a版本的api，调用CreateFile的时候

#ifdef UNICODE

#define CreateFile CreateFileW

#else

#define CreateFile CreateFileA

#endif // !UNICODE

# ListView显示图标

借鉴了c# 基本上是一样的。

首先前台先放一个listview控件和imagelist控件

点击imagelist控件，在它的iamges属性中添加图片，点击listview控件将它的SmallImageList属性设置为imageList

后台：

ListViewItem^ ls=gcnew ListViewItem();

listView1->Items->Add(ls);

for(int i=0;i< imageList1->Images->Count;i++)

{

//listView1->Items[i]->ImageIndex=i;

listView1->Items->Add(Convert::ToString(i),i);

/\* ls->ImageIndex=i;\*/

}

# Join函数

String类定义了一个Join（）函数，可用于将数组中存储的多个字符串连接成一个字符串，原来的字符串之间用分隔符分开。下面是将多个姓名连接成一个字符串的方法，各个姓名以逗号分开：

Array<StringY^>^ names={L”Jill”,L”Ted”,L”Mary”,L”Eve”,L”Bill”};

String^ separator(L”, “);

String^ joined=String::Join(separator,names);

执行这些语句之后，joined将引用字符串L”Jill,Ted,Mary,Eve,Bill”.在names数组中原来的各个字符串之间插入分隔符字符串。当然，分隔符字符串可以是任何我们喜欢的内容，如可以是L”and”,这样结果字符串将是L”Jill and Ted and Mary and Eve and Bill”.

# 对话框

String^ msg = "Do you want cancel burn process?";

String^ caption = "Warning!";

::DialogResult result = MessageBox::Show(this, msg, caption,

MessageBoxButtons::YesNo, MessageBoxIcon::Warning);

if (result == ::DialogResult::Yes) {

MessageBox::Show(this,"我点击了提交","提示",MessageBoxButtons::OK,MessageBoxIcon::Warning);

}

else if(result==::DialogResult::No)

{

this->Close();//关闭窗体

}

或者直接MessageBox::Show("我点击了提交");也行。

# 画笔

在.Net Framework里面画图主要是通过调用System::Drawing这个类库实现的。

在画图时，首先要创建一个System::Drawing::Graphics类型的对象，以后所有的画图动作都由这个对象来完成，创建这个对象的代码如下：

System::Drawing::Graphics^ formGraphics;

formGraphics = this->CreateGraphics();

然后还要创建一个画刷，这个画刷定义了画出来的图形的颜色。创建这个画刷的代码如下：

System::Drawing::SolidBrush^ myBrush = gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);

这两个对象都创建好后，画圆就很简单了，具体代码如下：

formGraphics->FillEllipse(myBrush, Rectangle(100, 100, 200, 200));

其中那个Rectangle(100, 100, 200, 200)矩形对象是定义圆的大小的。

画完图形后，最好把刚才建立的那两个对象删除，这样可以节约系统的资源。

完整的画圆代码如下：

private: System::Void DrawCircleButton\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Drawing::SolidBrush^ myBrush = gcnew System::Drawing::SolidBrush(System::Drawing::Color::Blue);

System::Drawing::Graphics^ formGraphics;

formGraphics = this->CreateGraphics();

formGraphics->FillEllipse(myBrush, Rectangle(100, 100, 200, 200));

formGraphics->DrawEllipse(Pen, Rectangle(50, 50, 200, 200));//如果想要画一个有边框的椭圆 就加上这个。

delete myBrush;

delete formGraphics;

}

# 日期和时间

可以在比如构造函数里面可以初始化时间和日期。如：

String^ mVersion=”1.0”;

String^ s =String::Format(“{0} v{1}<build {2}-{3}>”,Text,mVersion,\_\_DATE\_\_,\_\_TIME\_\_)

运行后：

frmMain v1.0<build Jun 27 2012-13:50:27>

text是获取当前窗体的text值，\_\_DATE\_\_是获取当前日期，\_\_TIME\_\_是获取当前时间

# List<String^>^

如果定义成 List<String^>^ mBinDirs;需要在.h文件中添加using namespace Collections::Generic;

# 获取路径

AppDomain::CurrentDomain->BaseDirectory运行后显示路径是：

D:\Program Files\Microsoft Visual Studio 10.0\Projects\test111\Debug

Directory::GetCurrentDirectory()运行后显示路径是：

D:\Program Files\Microsoft Visual Studio 10.0\Projects\test111\

# 获取端口号

array<String^>^ ports = SerialPort::GetPortNames();

# 程序退出

pplication.Exit()

方法是终止所有线程上的消息循环，一般情况下，无论在什么地方调用此方法，程序就能退出。

但是如果你在程序中加入了某些耗时甚至是死循环的线程，那么即使是消息循环终止，程序也依然不会结束。

比较温和比较合适的做法是结束消息循环之前，终止所有自己新建的线程。

System::Environment::Exit(0);

应用程序即强制退出。

# 错误验证控件errorProvider

首先从工具栏拖一个errorProvider组件。

如果验证textBox1控件的值范围不能是10-20之间 。则添加textBox1的Validating事件

后台代码：

System::Void aaa::textBox1\_Validating(System::Object^sender, System::ComponentModel::CancelEventArgs^ e)

{

if(Convert::ToInt32(textBox1->Text)>10 && Convert::ToInt32( textBox1->Text)<20)

{

errorProvider1->SetIconAlignment((Control^)sender,ErrorIconAlignment::MiddleRight);

errorProvider1->SetError((Control^)sender,"数字在10和20之间了，请更改!");

e->Cancel=true;

}

else

errorProvider1->SetError((Control^)sender,"");//如果没有错误，就调用SetError()方法，并把空字符串作为错误字符串，清除ErrorProvider。

}

# vs2010 快捷方式

Ctrl + i 渐进搜索

用F3/Shift+F3的组合在文档里向前向后搜索。

Ctrl + 减号 /向前定位 和 Shift + Ctrl + 减号 /向后定位