

## Delphi 编程：用流来读取 TXT 文件中的数据

<http://www.enet.com.cn/esoftware/> 2007 年 09 月 12 日 15:27 来源: eNet

硅谷动力 作者:xqi 字号: 小 | 大

【文章摘要】TXT 文本文件是一种以 ACSII 吗存储数据的文件。在 TXT 文件中只能够存储一些字符数据，所以这种文件的移植性和通用性很强，具有较高的易用性，并且被广泛应用，所以 Delphi 对该类型文件提供了较好的支持。目前在 Internet 上最为流行的 XML 文件也是一种文本文件。

【eNet 硅谷动力专稿】TXT 文本文件是一种以 ACSII 吗存储数据的文件。在 TXT 文件中只能够存储一些字符数据，所以这种文件的移植性和通用性很强，具有较高的易用性，并且被广泛应用，所以 Delphi 对该类型文件提供了较好的支持。目前在 Internet 上最为流行的 XML 文件也是一种文本文件。TXT 文本文件的操作非常简单，具体操作过程如下：

1. 首先要声明一个 TextFile 类型的的文件指针。如：var MyFile:TextFile。
2. 然后使用 AssignFile 方法将声明的文件指针与外部 TXT 文件相关联。AssignFile (FileName) ; FileName 既可以是全路径名,也可以仅是文件名。对于后者系统将在当前目录下查找。
3. 初始化读写有三种方式：Reset 为读打开文件并把文件指针移动到文件首；Rewrite 为写创建一个新文件；Append 为写打开存在的文件并把文件指针定位在文件尾。
4. 读取或写入文本。Readln 方法能够读取一行文本，writeln 方法能向文件写入一行文本。
5. 最后用 Closefile 方法关闭已打开的文件。

在这里我们要介绍的就是如何用流来读取 TXT 文件中的数据，并且在指定显示区域显示出来。首先让我们通过一个简单范例来了解 ReadBuffer 方法，让我们来读取指定 TXT 文件的前 10 个字符。

```
procedure TForm1.Button1Click(Sender: TObject);

var

MyFile:TMemoryStream;

Filebuf: array[1..10] of char; //这里声明的是静态数组
```

```

a:string;

Begin

If opendialog1.Execute then

begin

MyFile:=TMemoryStream.Create;//创建流

MyFile.LoadFromFile(opendialog1.FileName);

MyFile.ReadBuffer(Filebuf,10);//读取 TXT 文件前 10 个字符

a:= strpas(Filebuf) //将数组转化成为字符串

Form1.canvas.Textout(0,0,a); //在显示区域上显示字符串

FreeAndNil(MyFile);//释放流

end;

end;

```

以上范例在窗体 Form1 上显示的字符串就是打开 TXT 文本的前 10 个字符，但是如何得到文本的所有数据呢？这里就需要用到动态数组。动态数组在流中的应用是个难点，因为 Delphi 中的静态数组是在运行前就已经将内存空间分配好，所以它的变量地址就是数组的第一维地址，即没有描述部份，故它的 sizeof 为 1 个字节，而动态数组是在运行期间动态分配一个内存块，所以它的变量地址部分需要一个描述部分，故它的 sizeof 为四个字节，用于存放描述表，所以在用 Readbuffer 方法时，需要采用数组的第一维地址为起始地址，才不会导致内存溢出。下面这个范例就是用流来读取 TXT 文件中的全部数据。

```

procedure TForm1.Button1Click(Sender: TObject);

var

MyFile:TMemoryStream;

```

```

FileBuf: array of pchar; //这里声明的是动态数组

iLen: Int64;

Begin

If opendialog1.Execute then

begin

iLen:=0;

MyFile:=TMemoryStream.Create;

MyFile.LoadFromFile(opendialog1.FileName);

iLen:=MyFile.Size;//获得指定 TXT 文件的大小

SetLength(FileBuf, iLen);//设置动态数组的长度为 TXT 文件的大小

MyFile.ReadBuffer(FileBuf[0], iLen);//读取 TXT 文件全部数据

Form1.canvas.Textout(0, 0, string(FileBuf)); //在显示区域上显示字符串;

FreeAndNil(MyFile);

end;

end;

```

以上的范例可在窗体 Form1 上显示指定 TXT 文件的全部数据。

现在让我们来研究一下如何从 TXT 文本中的指定位置读取一定量的数据, 这里用到的函数就是 seek (offset:integer, origin:word) integer; 它的参数所代表的意义: offset 是偏移量;

而 origin 是计算方式, 下面三个就是 origin 的值, soFromBeginning 是从文件头开始计算, soFromCurrent 是从当前位置开始计算, 而 soFromEnd Offset 是从最后位置开始

计算。

下面这个范例就是从 TXT 文本中的指定位置取一定量的数据

```
procedure TForm1.Button1Click(Sender: TObject);

var

MyFile:TMemoryStream;

FileBuf: array of pchar; //这里声明的是动态数组

iLen: Int64;

Begin

If opendialog1.Execute then
begin
iLen:=0;

MyFile:=TMemoryStream.Create;
MyFile.LoadFromFile(opendialog1.FileName);

iLen:=MyFile.Size;

SetLength(FileBuf, 1024); //设置动态数组的长度;

Myfile.Seek(1024, soFromBeginning); //从文件头开始计算到 1024 个字节处

MyFile.ReadBuffer(FileBuf[0], 1024); //从 seek 设置的当前位置往后读取 1024 字节

Form1.canvas.Textout(0, 0, string(FileBuf)); //在窗体上显示;

FreeAndNil(MyFile);
```

end;

end

