

块。道头块一般包括道头标识,道尺寸,数据块长度,每道样点数,数据段格式码,途径文件名,道号,采样率,延迟时间,低截,测线号,炮点位置,检波器位置,叠加次数,作图幅度等内容。

道头块的第0-1字节为一个十六进制无符号整型量4422H作为道头标识。第2-3字节为道头尺寸,是一个无符号的整型量。第4-7字节为数据段长度,为样点数/道*4字节。第8-11字节为每道样点数,是一个无符号的长整型量。第12字节为数据段存储数据的格式码,1H表示16位整数,2H表示32位整数,3H表示20位浮点数,4H表示32位浮点数,5H表示64位浮点数。第13-31字节为SEG保留内容,以备扩充。第32字节直至字节内容为0,均为ASCII字符串,由于篇幅所限,此处不一一描述。每道紧跟道头后的是数据段,具体数据内容由此段读出。图3所示为由WinHex显示出的SEG-2道头块ASCII字符串。

```

20 00 00 00 00 00 00 00 11 00 43 44 50 5F 4E 55 ..... CDP_NU
4D 42 45 52 20 37 35 30 00 0E 00 43 44 50 5F 54 MBER 750...CRP_T
52 41 43 45 20 31 00 0A 00 44 45 4C 41 59 20 30 RACE 1...DELAY 0
00 1E 00 52 45 43 45 49 56 45 52 5F 47 45 4F 4D ...RECEIVER_GEOM
45 54 52 59 20 31 2E 37 33 65 2B 30 30 36 00 1E ETRY 1.73e+006..
00 52 45 43 45 49 56 45 52 5F 4C 4F 43 41 54 49 .RECEIVER_LOCATI
4F 4E 20 31 2E 37 33 65 2 10 18 00 53 ON 1.73e+006...S
41 4D 50 4C 45 5F 49 4E 5 ASCII字符串 1 4C 20 30 AMPLE_INTERVAL/G
2E 30 30 34 00 1B 00 53 4 13 45 51 55 .004...SHOT_SEQH
45 4E 43 45 5F 4E 55 4D 42 45 52 20 34 33 36 00 ENCE_NUMBER 436.
1E 00 53 4F 55 52 43 45 5F 4C 4F 43 41 54 49 4F ..SOURCE_LOCATIO
4E 20 32 2E 30 32 37 35 65 2B 30 30 36 00 00 00 N 2.0275e+006...

```

图3 由WinHex显示出的SEG-2道头块ASCII字符串

2 VC++建立动态链接库(DLL)

打开VC++,新建一个MFC AppWizard(dll)工程,命名为Seg2。建立C/C++ Header File,命名为Seg2_Read.h。建立C/C++ Source File,命名为Seg2_Read.cpp。两个文件的代码及解释如下。

在Seg2_Read.h文件中键入:

```
extern "C" __declspec(dllexport) char *outputstring();
//向调用程序声明可被调用的函数
```

在Seg2_Read.cpp文件中键入:

```
#include "StdAfx.h"
#include "seg2_read.h"
#include <stdio.h>
unsigned short int stringlength; //数据文件描述块结构体变量
struct seg2_hd32
{ short blockid;
  short revnum;
  unsigned short pointerbytecount;
  unsigned short tracenum;
  char stringtermcount;
  char stringterm1;
  char stringterm2;
  char linetermcount;
  char lineterm1;
  char lineterm2;
  char reserved[18];
} seg2_hd32;
unsigned int tracepointer[100];
```

```
//数据文件描述块ASCII字符串变量
char dscriptstring[100];
void read_file() //读取文件函数
{ FILE *fseg2;
  fseg2=fopen(".\\data_seg2.sgy","rb");
  fread(&seg2_hd32,32,1,fseg2);
  fread(tracepointer,4,seg2_hd32.tracenum,fseg2);
  fread(&stringlength,2,1,fseg2);
  //读取数据文件描述块ASCII字符串
  unsigned short int i=0;
  while(0!=stringlength)
  { unsigned short int j=1;
    while(j<=(stringlength-2))
    { fread(&dscriptstring[i],1,1,fseg2);
      j=j+1;
      i=i+1;
    }
    fread(&stringlength,2,1,fseg2);
  }
  fclose(fseg2);
} //可被调用的函数
extern "C" __declspec(dllexport) char *outputstring()
{ read_file();
  return dscriptstring; //返回ASCII字符串数据指针
}
```

将数据文件data_seg2.sgy放置于工程目录下,点击执行,在工程目录Debug下生成了Seg2.DLL。我们将在下一步使用它。

3 Delphi调用动态链接库(DLL)中的函数

在Delphi中新建工程Seg2_Read,布置面板如图4所示。由于篇幅有限,我们使用隐式调用。

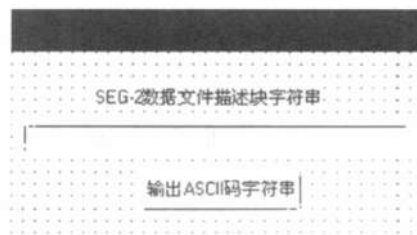


图4 Delphi面板布置

新建一个单元,命名为opstring.pas,分别在interface和implementation下键入以下代码:

```
interface
function outputstring( ): pchar; stdcall;
implementation
function outputstring;external 'seg2.DLL' name 'outputstring';
//调入DLL
在Seg2_Read.pas的implimentation下键入以下代码:
uses
opstring;
procedure TForm1.Button1Click(Sender: TObject);
```

(下转第45页)

```

{ name=document.form1.name.value;
  if (chkspc(name)==0||fucCheckLength(name)<4)
  { alert("请填写您的姓名,姓名由1到4个字符组成。");
    document.form1.name.focus(); return false;
  }
  pwd=document.form1.pwd.value;
  if (fucCheckLength(pwd)<6||fucCheckLength(pwd)>
    12||fucPWDchk(pwd)==0)
  { alert("密码由6到12位字母或数字组成。");
    document.form1.pwd.focus(); return false;
  }
  email=document.form1.email.value;
  if ((chkspc(email)==0) || (chkemail(email)==0)||
    (fucCheckLength(email)>40))
  { alert ("请填写正确的e-mail地址。");
    document.form1.email.focus(); return false;
  }
  postcode=document.form1.postcode.value;
  if (chkspc(postcode)==0||fucCheckLength(postcode)>15)
    ||fucCheckNUM(postcode)==0)
  { alert ("请您填写正确的邮政编码,它由数字组成。");
    document.form1.postcode.focus(); return false;
  }
  usephone=document.form1.usephone.value;
  if (usephone==" || chkspc(usephone)==0 || fucCheckLength
    (usephone)>30||fucCheckTEL(usephone)==0)
  { alert("您的常用电话号码填写有误。");
    document.form1.usephone.focus(); return false;
  }
}

```

```

date=document.form1.date.value;
if (chkspc(date)==0||chkdate(date)==0)
{ alert ("请填写您的订货日期,格式如2010-1-8。");
  document.form1.date.focus(); return false;
}
}
</ script>

```

3 结束语

本文提出了一种基于JavaScript 的验证表单数据的通用方案,并给出了相应的验证程序代码。该方案可以对用户信息进行详细验证。该方案将验证机制对应在一个外部脚本文件 Client Event Handlers.js 中,通过调用这个文件实现代码复用。该方案能够在多种环境下应用,其主要技术特点有:①跨平台、跨浏览器运行;②适合多种表单网页动态验证数据。应用范例和实际运行表明,该方案简洁实用,较好地满足了通用性的应用需求。

参考文献:

- [1] Tom Negrino, Dori Smith 著,陈创既等译. JavaScript 基础教程(第7版)[M]. 人民邮电出版社, 2009.
- [2] Michael Moncur 著,王军译. JavaScript 入门经典(第4版)[M]. 人民邮电出版社, 2008.
- [3] 熊圣芳, 贺智明, 王俊. 基于 JavaScript 和 DOM 的动态表单设计及数据提交[J]. 江西理工大学学报, 2009. 2: 31~33
- [4] 张彦芳, 谢虹. 基于 JavaScript 的客户端表单验证[J]. 生产一线, 2009: 33
- [5] 徐卫英. 《脚本语言与动态网页设计》课程教学案例的设计[J]. 苏州大学学报, 2008. 12: 79~81



(上接第 42 页)

```

begin
  edit1.text=outputstring(); //调用动态链接库中的函数
end;

```

将前面已生成的 Seg2.DLL 及数据文件 data_seg2.sgy 拷贝到工程目录下, 点击运行, 然后在运行窗口上单击“输出 ASCII 字符串”按钮, 文本框显示如图 5 所示。可见, 与图 2 中 WinHex 显示出的 SEG-2 文件描述块 ASCII 字符串相符合。

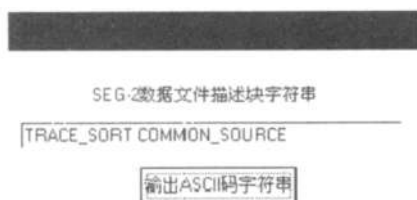


图5 Delphi调用Seg2.DLL读出的SEG-2文件描述块ASCII字符串

至此, 我们介绍了使用 Delphi 调用 VC 生成的动态链接库 (DLL) 来实现数据文件描述块 ASCII 字符串读取的步骤, 这种方法比纯 Delphi 或 C 语言编程省时, 省力。

4 结束语

本文详细介绍了使用 Delphi 调用 VC++ 生成的动态链接库 (DLL) 来实现地质和工程数据文件描述块 ASCII 字符串读取的方法和步骤。只要稍作修改就可以通过这样的方法读取文件中其他所有的地质和工程数据资料, 进行下一步的计算和管理工作。虽然, 这里只以动态链接库封装数据读取函数的实例, 但是, Delphi 也可以通过这样的方式调用其他的函数。这种方法不仅发挥了 C 语言底层编程计算大量地质和工程数据速度快的优势, 还利用了 Delphi 简单、易学、易用的界面图形编程特点。

参考文献:

- [1] 牛汉民. Delphi7 开发基础教程[M]. 科学出版社, 2005.
- [2] 肖庆航. Delphi7 数据库开发教程[M]. 清华大学出版社, 2004.
- [3] 侯德志. 深入浅出 MFC[M]. 华中科技大学出版社, 2001.
- [4] 张慧龙, 刘君宝. 面波法的数据采集处理系统及其应用实例[J]. 西部探矿工程, 2007. 12.
- [5] 徐维秀, 段卫星. 表层结构调查资料的数字化处理及应用[J]. 物探化探计算技术, 2005. 5.

