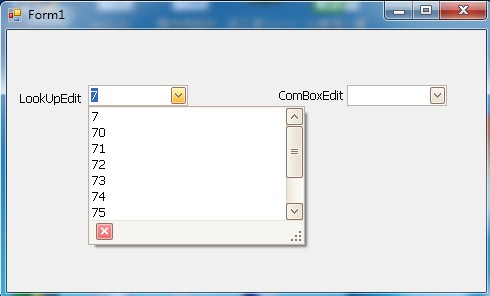
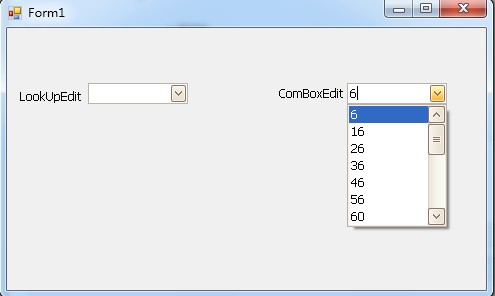
这几天老大叫我修改一下项目中LookUpEdit控件的下拉选择功能，实现模糊检索数据，经过一番研究，发现LookUpEdit控件只能实现部分模糊检索数据功能，也就是它的检索索引是从第一位开始，也就是模糊检索数据的时候，用户必须从头开始检索，这样给用户不友好，如果只知道其中的某些关键字则实现不了。如下图：



查询了资料后，发现LookUpEdit不支持自由输入，所以只好用它的兄弟控件ComBoxEdit控件。ComBoxEdit控件能够不受数据源的影响而自由输入值，现在就来说下我的处理思路： 根据用户输入的值，对ComBoxEdit的数据源做过滤，然后重新绑定到控件，废话不多说了，直接上代码吧。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93 | using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Windows.Forms;  using DevExpress.XtraEditors.Controls;    namespace WindowsFormsApplication1  {      public partial class Form1 : Form      {          public Form1()          {              InitializeComponent();          }          DataTable dt = new DataTable();          private void Form1\_Load(object sender, EventArgs e)          {              try              {                    BindData();              }              catch (Exception)              {                    //TODO              }            }         /// <summary>         /// 给全局dt复制，这里可以是sql语句，得到table后循环table绑定数据到控件         /// </summary>          private void BindData()          {              dt.Columns.Add(new DataColumn("value", typeof(string)));              for (int i = 0; i < 100; i++)              {                  DataRow dr = dt.NewRow();                  dr["value"] = i.ToString();                  dt.Rows.Add(dr);                  this.comboBoxEdit1.Properties.Items.Add(i);//绑定数据到控件              }              this.lookUpEdit1.Properties.DataSource = dt;              this.lookUpEdit1.Properties.ValueMember = "value";              this.lookUpEdit1.Properties.DisplayMember = "value";              this.lookUpEdit1.Properties.ShowHeader = false;                }              private void comboBoxEdit1\_KeyUp(object sender, KeyEventArgs e)          {              try              {                  string str = comboBoxEdit1.Text.ToString();                  //if (str == "")                  //    return;                  comboBoxEdit1.Properties.Items.Clear();//无论有没有过滤，都要清空原来的值                  string s = "value like '%" + str + "%'";                  DataView v = dt.DefaultView;                  v.RowFilter = s;                    DataTable dtt = v.ToTable();                  if (dtt.Rows.Count > 0)//如果输入的值过滤后有满足的值，则加载满足条件的值,否则加载全部                  {                      for (int i = 0; i < dtt.Rows.Count; i++)                      {                          this.comboBoxEdit1.Properties.Items.Add(dtt.Rows[i]["value"].ToString());                      }                  }                  else                  {                      for (int i = 0; i < dt.Rows.Count; i++)                      {                          comboBoxEdit1.Properties.Items.Add(dt.Rows[i]["value"].ToString());                      }                  }              }              catch (Exception)              {                    //TODO              }            }      }  } |

　　运行效果如下图：



小弟做的比较粗糙，希望大虾们多多指点。