

设置并加密App.config中数据库连接字符串 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/462/2021_2022__E8_AE_BE_E7_BD_AE_E5_B9_B6_E5_c102_462003.htm

App.config中的数据库连接字符串，是在Settings.settings同步过来的。后者在设计时支持数据集DataSet的设计，但运行时不能更改，可以更改App.config的连接字符串，但无法加密、解密。造成的结果是数据库连接字符串，要么固定不变，要么以明文显示，极不安全。多次尝试后，成功的解决这些问题。我的经验：1. Settings.settings中的设置和App.config中的设置会相互更新，彼此同步。前者用于设计支持，后者用于运行时。可以手工阻止两者同步，使两者设置不一样，如果没有发布App.config，则全部采用Settings.settings的设置；否则在运行时App.config会覆盖Settings.settings的设置。2. Settings.settings中的Application范围设置只在设计时更改，是只读的属性，User范围设置可以在运行时读写。应用程序作用域设置与用户作用域设置之间的重要区别是，用户作用域设置在运行时为读/写，并且可在代码中对其值进行更改和保存。应用程序作用域设置在运行时为只读。虽然可以读取，但是不能对其进行写入。具有应用程序作用域的设置只能在设计时或通过手动修改设置文件进行更改。3. Settings中的只有“连接字符串”设置才能被数据集DataSet设计时支持，而“连接字符串”设置只能是Application范围设置，是只读的属性。解决思路：

将Settings.settings的设计时代码文件Settings.Designer.cs内容复制到Settings.cs中，删除Settings.Designer.cs，更改部分代码。连接字符串在Settings.cs中缺省设置为明码，在App.config中为

加密码。代码如下，DESEncrypt为静态加密函数，DESDecrypt为静态解密函数：重写Settings代码 1 using System.Xml. 2 namespace A.Properties 3{ 4 5 [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()] 6 [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator", "8.0.0.0")] 7 internal sealed class Settings : global::System.Configuration.ApplicationSettingsBase 8 { 9 10 private static Settings defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings()))).11 12 public static Settings Default13 {14 get15 {16 return defaultInstance.17 }18 }19 20 [global::System.Configuration.ApplicationScopedSettingAttribute()]21 [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]22 [global::System.Configuration.SpecialSettingAttribute(global::System.Configuration.SpecialSetting.ConnectionString)]23 [global::System.Configuration.DefaultSettingValueAttribute ("Data Source=(local).Initial Catalog=Test.Persist Security Info= True.UserID=sa.Password=")]24 public string ConnectionString125 {26 get27 {28 return Encrypt.DESDecrypt(this["ConnectionString1"].ToString()).29 }30 set31 {32 base["ConnectionString1"] = Encrypt.DESEncrypt(value).33

```

this.SetKeyValue("A.Properties.Settings.ConnectionString1",
Encrypt.DESEncrypt(value)).34 }35 }363738 /**//// 39 /// 保存设
置40 /// 41 /// 42 /// 43 private void SetKeyValue(string AppKey,
string AppValue)44 {45 XmlDocument xDoc = new
XmlDocument().46
xDoc.Load(System.Windows.Forms.Application.ExecutablePath
".config").4748 XmlNode xNode.49 XmlElement xElem1.50
XmlElement xElem2.5152 xNode =
xDoc.SelectSingleNode("//connectionStrings").5354 xElem1 =
(XmlElement)xNode.SelectSingleNode("//add[@name=" AppKey
]").55 if (xElem1 != null) xElem1.SetAttribute("connectionString",
AppValue).56 else57 {58 xElem2 = xDoc.CreateElement("add").59
xElem2.SetAttribute("name", AppKey).60
xElem2.SetAttribute("connectionString", AppValue).61
xNode.AppendChild(xElem2).62 }63
xDoc.Save(System.Windows.Forms.Application.ExecutablePath
".config").64 }656667 }68}69 默认情况下，新的连接字符串设置
只在重启时生效。但可以操作数据前，加上一条更改Dataset
中Adapter的连接字符串的语句，就可以不用重启，代码如下
：运行更改数据库连接 1 private void
toolStripButton1_Click(object sender, EventArgs e) 2 { 3
A.Properties.Settings.Default.ConnectionString1 = @"Data Source=
(local).Initial Catalog=Test2.Persist Security Info=True.User ID=sa".
4 5 this.toolStripTextBox1.Text = A.Properties.Settings.Default.
ConnectionString1. 6
this.Test1TableAdapter.Connection.ConnectionString =

```

A.Properties.Settings.Default.ConnectionString1. 7

MessageBox.Show(A.Properties.Settings.Default.ConnectionString1
) . 8 this.Test1TableAdapter.Fill(this.dataSet1.Test1). 910 } 100Test

下载频道开通，各类考试题目直接下载。详细请访问

www.100test.com