

Anycell Report(AC Report)中国式报表工具

在 VFP 中的应用

作者：xinjie QQ:411618689

一 关于本文

AC Report 的历史、功能简介，在本文所附的《Anycell Report 应用开发手册》中有详细的介绍，这里需要重复说明的一点：在 AC Report 的诸多版本中，从 2.1 版开始，该报表工具由一个 ActiveX 控件，经重新规划和改进，以 COM + ActiveX 方式予以提供。

本文将以 1.84 版和 3.2.12 版为基准，描述 AC Report 在 VFP 中的应用方法。

二 开发环境

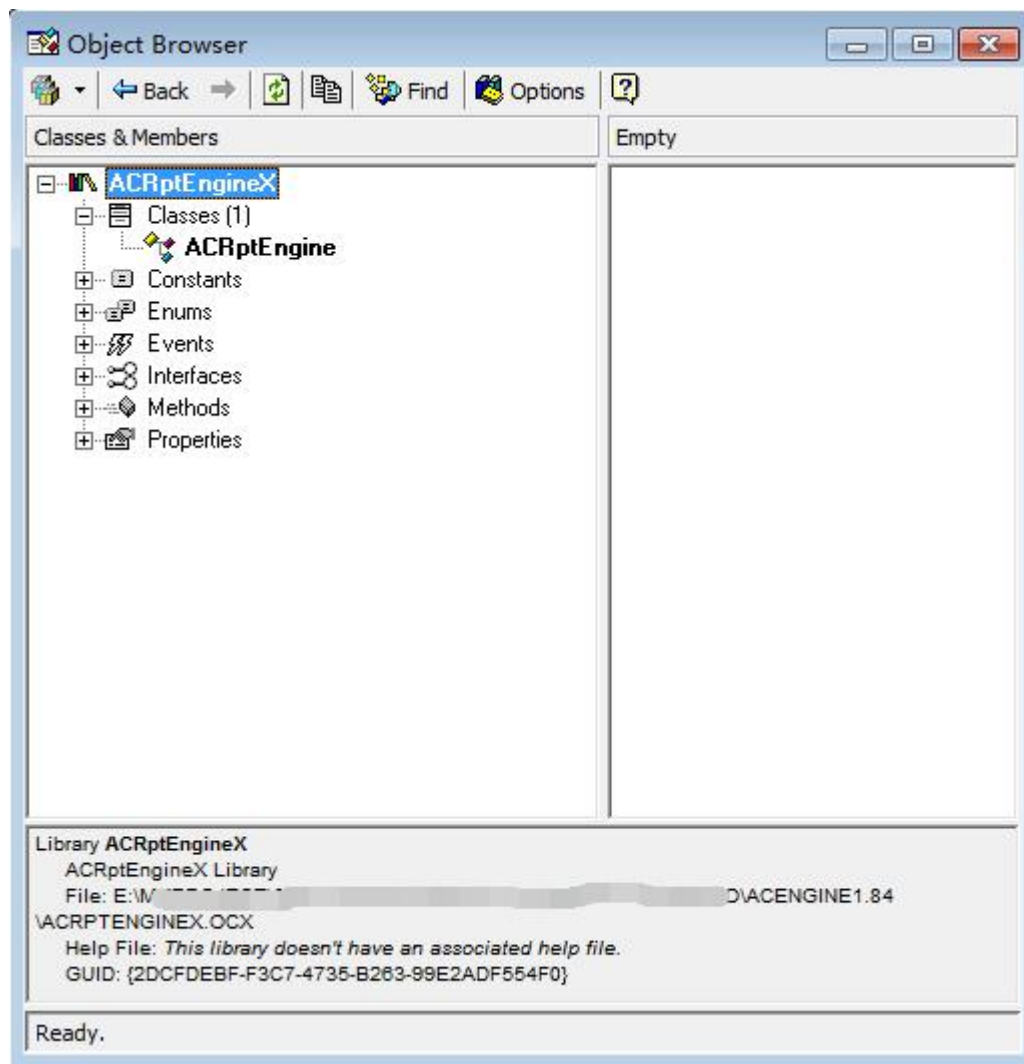
操作系统：Window 7 专业版（32 位）

编程语言：Visual Foxpro 9 （7423 版）

三 创建 AC Report 实例的基本方法

1 2.1 之前的版本

我们使用 Object Browser(对象浏览器)(菜单 Tools-Object Browser)先观察一下它的调用方法：



由此可知，可以使用以下语句来创建它的实例：

```
Private loACReport  
Local loACReport  
  
m.loACReport = Createobject([ACRptEngineX.ACRptEngine])  
m.loACReport.Init
```

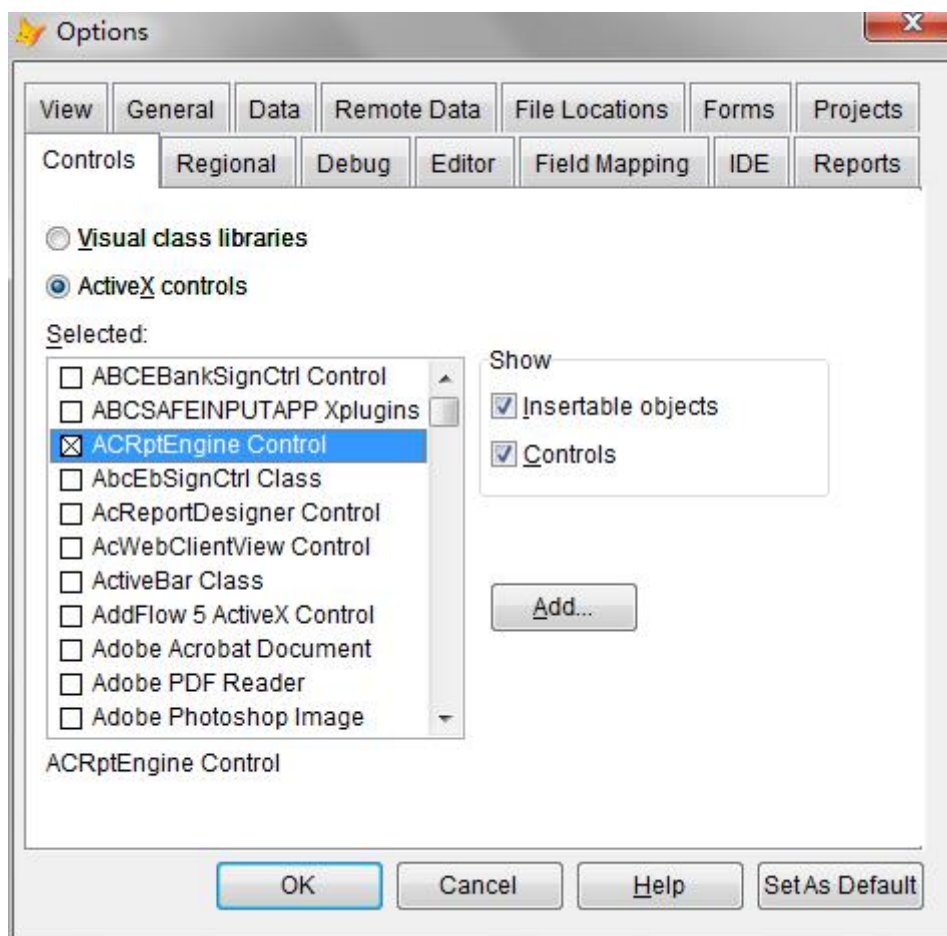
当然，作为一个 ActiveX 控件，你也可以通过拖拽的方式将它添加到一个容器中。仅仅需要按照下面的步骤操作即可：

第一步：通过菜单 Tools-Options... 打开“Options”对话框，并单击对话框的“Controls”标签

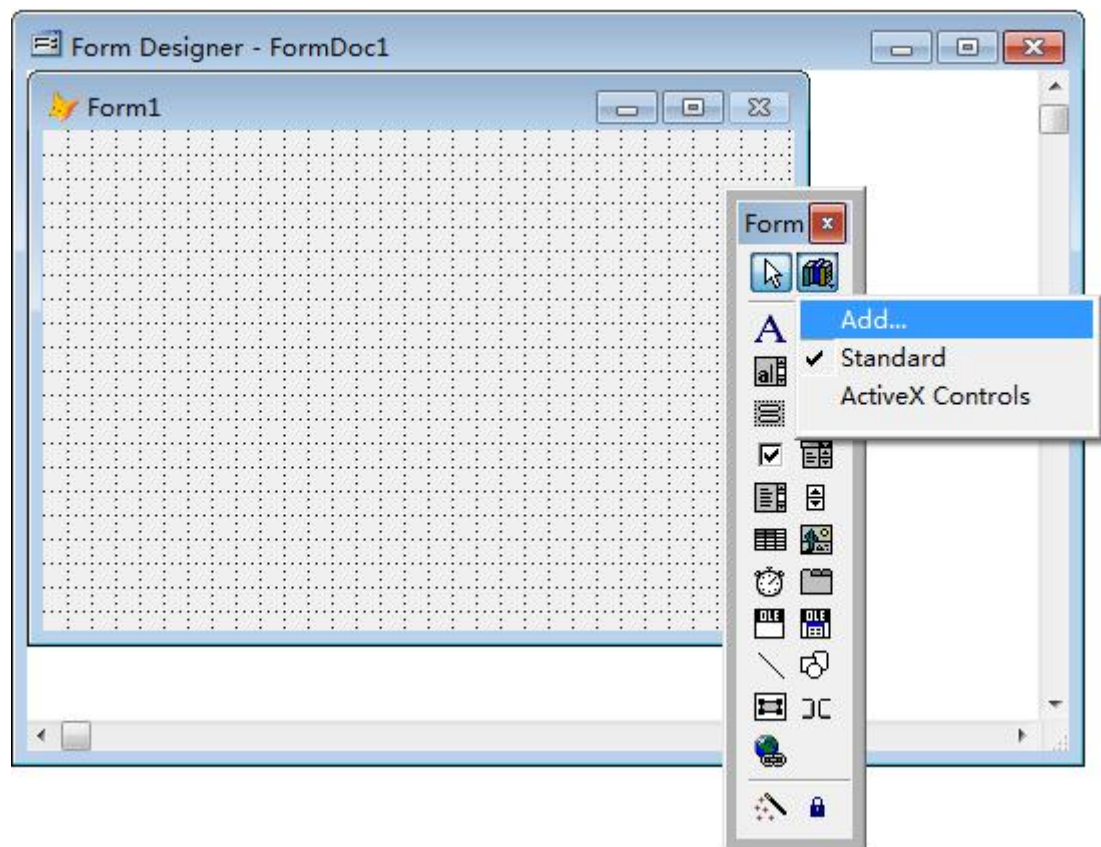
第二步： 通过点击“Add...”按钮注册 OCX 控件

第三步：选择 ACRptEngine

第四步：点击“OK”或者“Set As Default”按钮

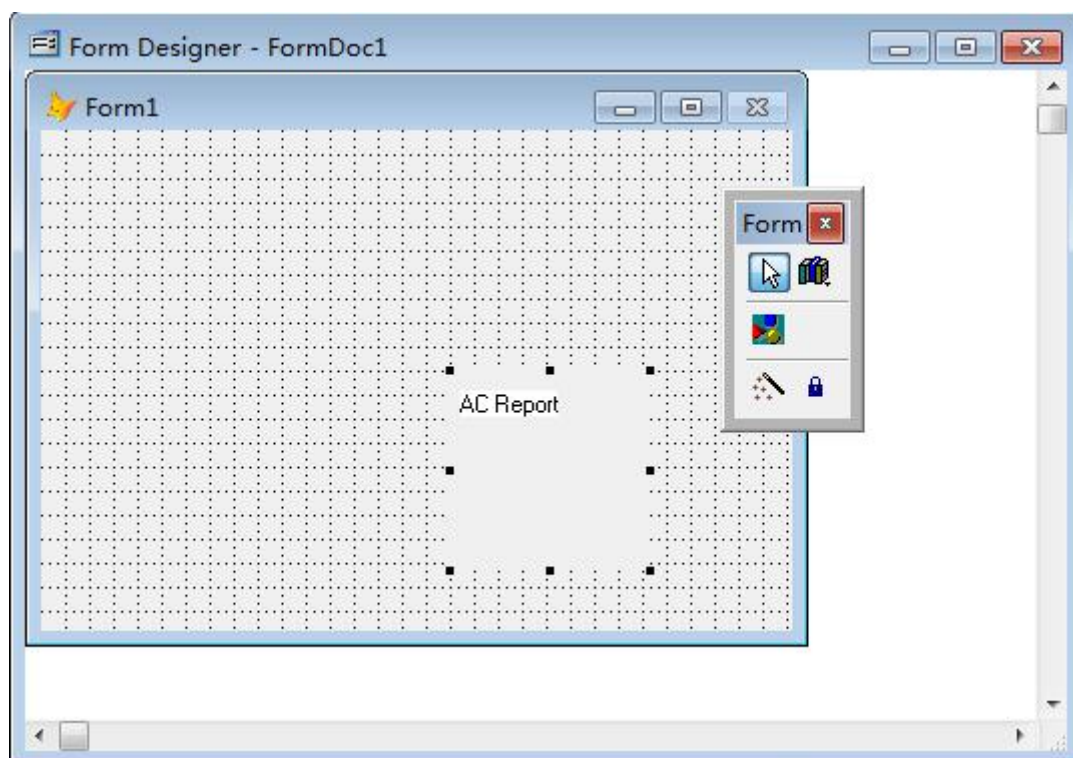


第五步：打开表单或者类设计器，启动“表单控件工具栏”



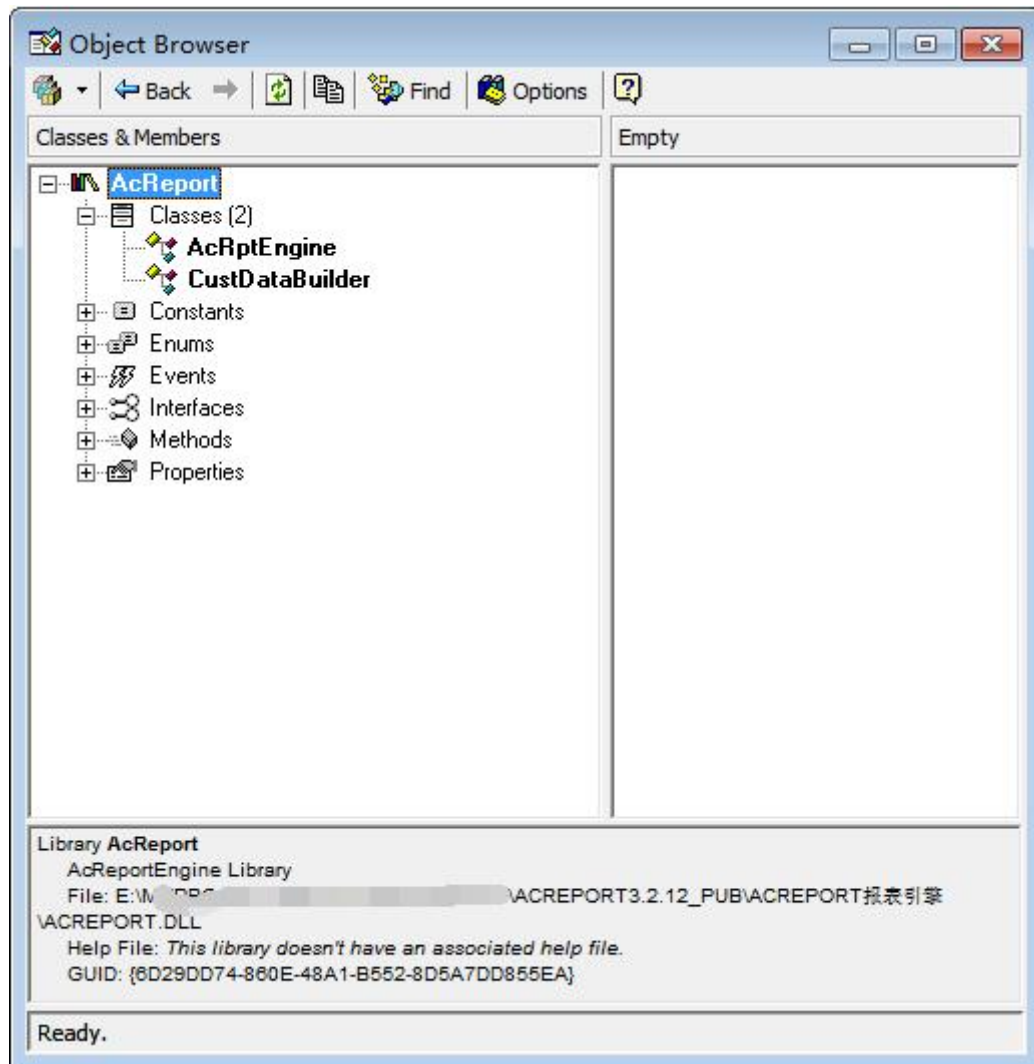
第六步：选择“ActiveX Controls”菜单项

第七步：点击 ACRptEngine Control 控件，并单击设计器的表单



2 2.1 及其之后的版本

由于作者对该工具做了重新规划设计，所以，2.1 及其之后的版本，通常情况下，将只能以代码的方式来创建实例：



`Private` `loACReport`

`Local` `loACReport`

```
m.loACReport = Createobject([AcReport.AcRptEngine])
```

```
m.loACReport.Init
```

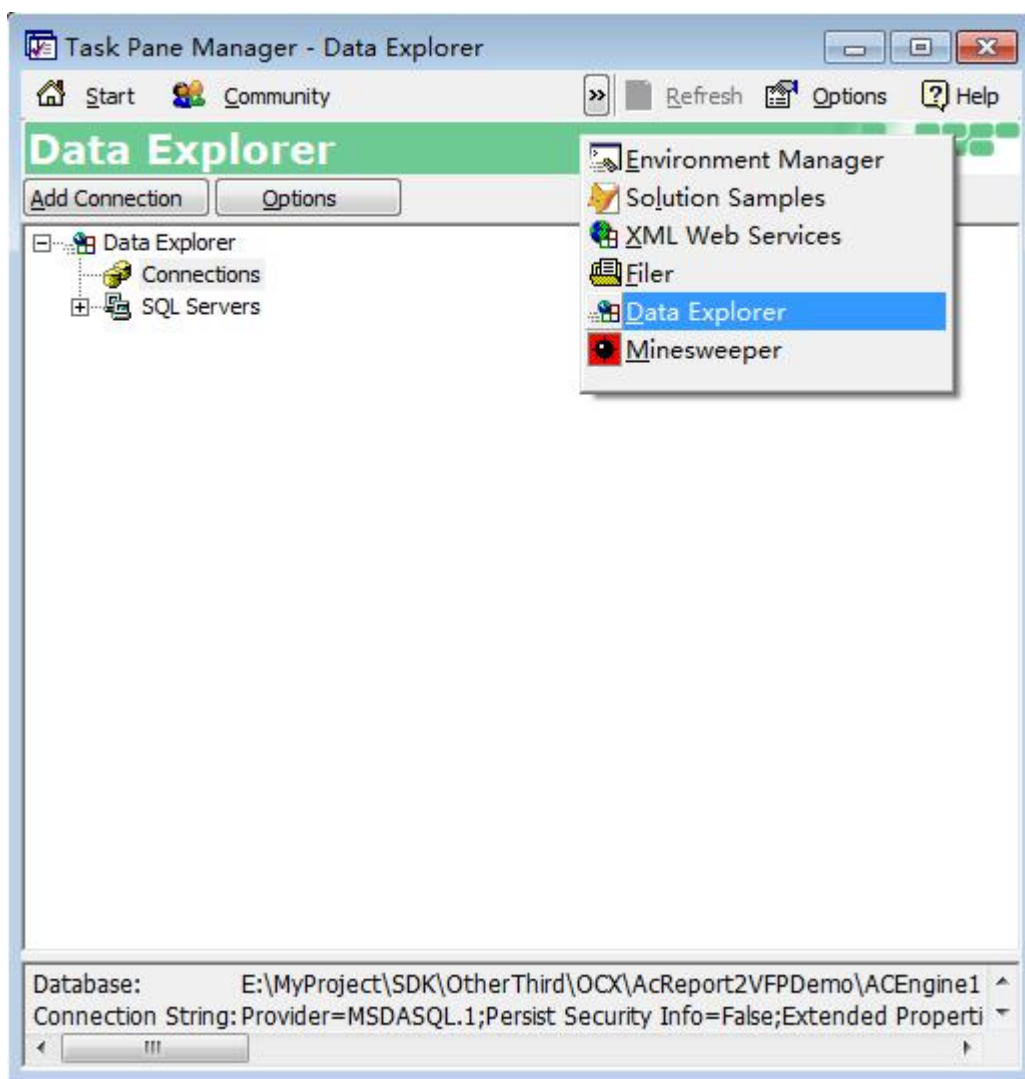
这里需要注意的是，2.1 及其之后的版本，它的 `Library` 与之前的版本是不同的！

四 AC Report 的基本使用方法

通过《Anycell Report 应用开发手册》，我们得知该控件数据源只能是 ADO 数据集。对于大多数 Foxer 来说，这个东西太过于陌生。但是，这并不表示 VFP 无法使用它。

我们先来做做热身运动：)

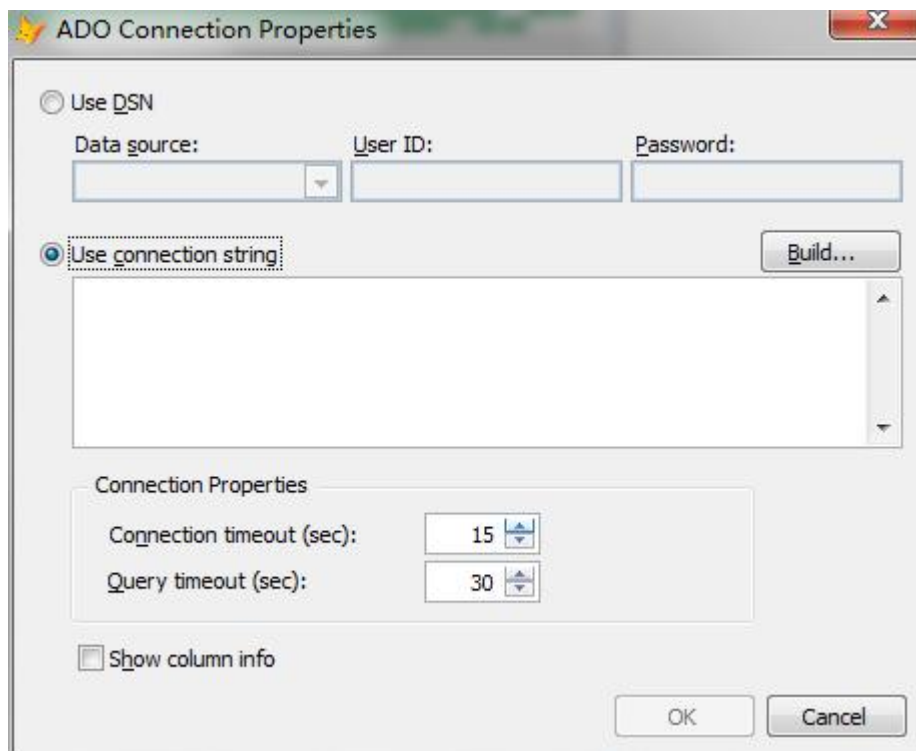
通过菜单 Tools-Task Pane 打开任务面板，然后大多数情况下，你看到的是如下图所示：



选择 Connections ，然后点击“Add Connection”按钮，会弹出如下对话框。
在该对话框中，选择“ADO Connection”并单击“OK”按钮



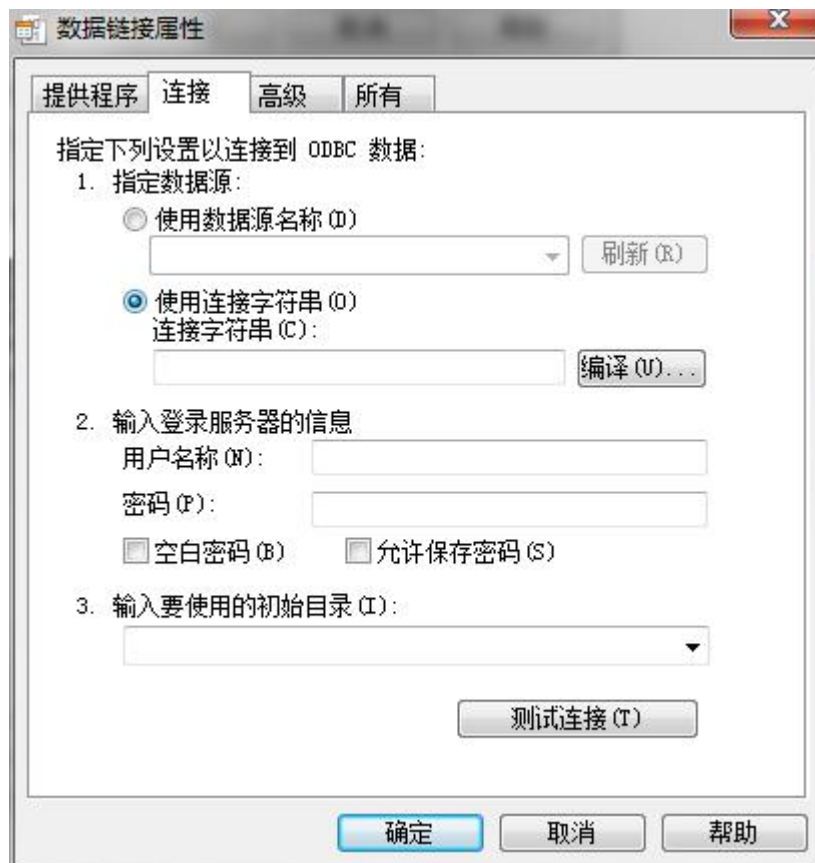
这样，就会打开一个 ADO Connection Properties 对话框：



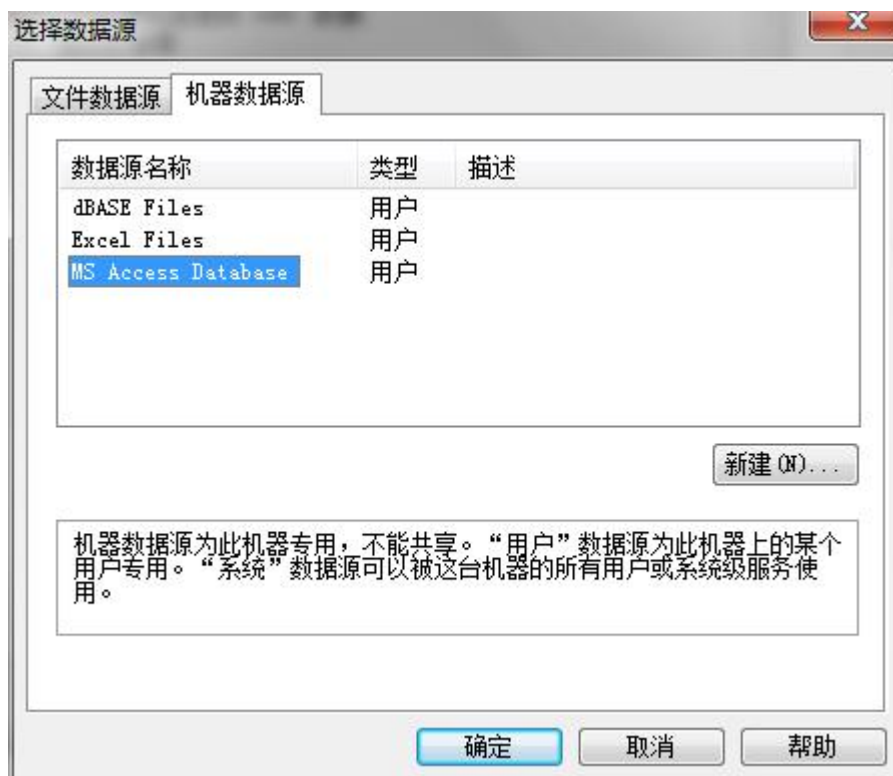
选择“Use connection string”，并单击“Build...”按钮，打开“数据连接属性”：



在其中选择一个可以连接到 ACCESS 数据库的驱动，我这里选择的是“Microsoft OLE DB Provider for ODBC Drivers”，然后单击“下一步”按钮：



选择“使用连接字符串”，并点击“编译...”按钮，会弹出如下对话框：



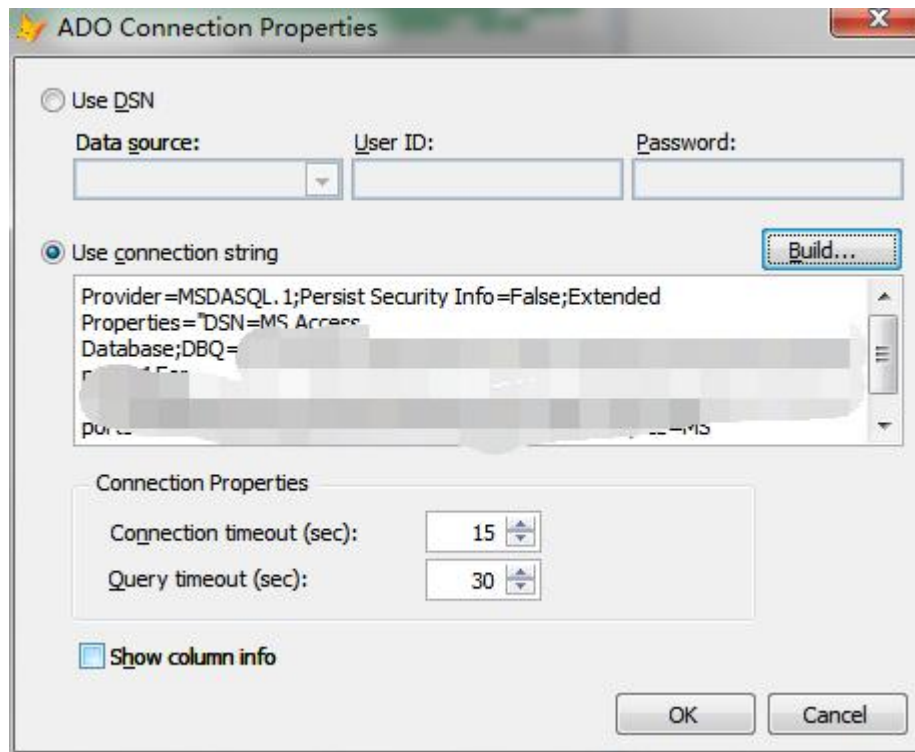
在这里，选择“MS Access Database”并点击“确认”按钮



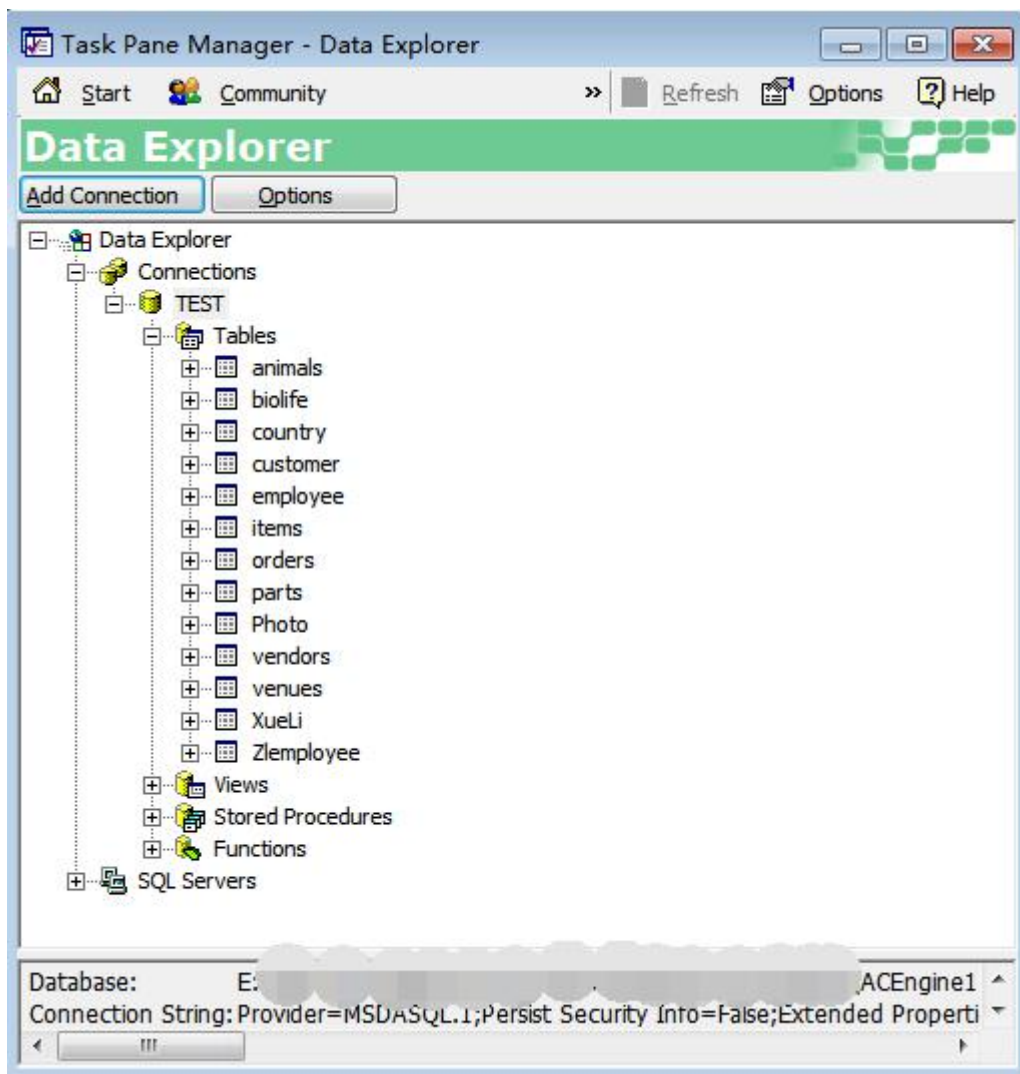
点击“数据库...”按钮，



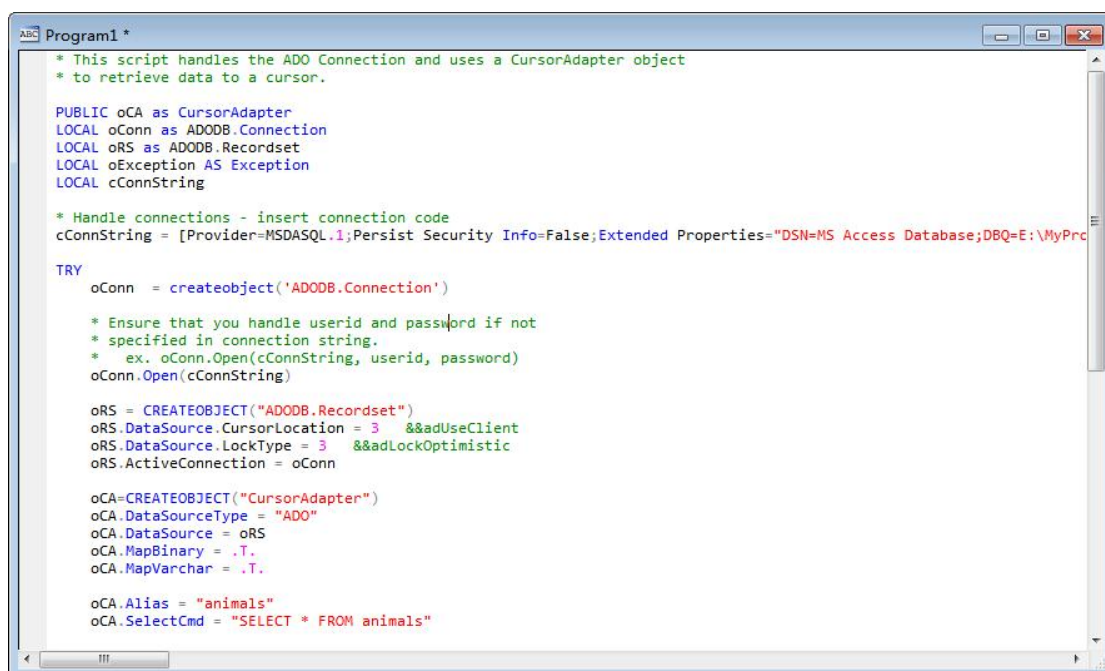
选定文件，并单击“确定”按钮，并依次关闭“登录”和“数据链接属性”，返回到 VFP 的 ADO Connection Properties 对话框，你看到的应该是类似下面截图的结果：



此时，单击“OK”按钮，返回到任务面板：



然后，新建一个 PRG，并拖动某一个表到编辑窗口，你会得到如下的结果：



这就是一个标准的使用 VFP 通过 ADO 连接来获取 ADO 数据集的方法。当然，现在看到的代码中，是使用了 CA 来进行数据操作，这一点和本文无关，暂时不用关注。

在这段代码中，你会看到一个像 ODBC 连接字符串一样的东西，并且，创建了一个 oConn ADO 连接对象，通过对象的 Open 方法，建立连接。在连接建立之后，使用 CREATEOBJECT("ADODB.Recordset") 创建了一个 ADO 记录集。

OK，到这里，热身结束，书归正传。那么在 VFP 中使用 AC Report 时，我们应该怎么写代码呢？首先，我们需要创建一个 ADO 连接对象，然后使用它建立和远程数据库的连接，并且通过调用连接对象的 execute 方法获得一个 ADO 记录集，然后通过调用 AC Report 对象的 ADDDataSet() 方法将 ADO 记录集“添加到对象中”，使得 AC Report 对象可以顺利使用数据源。一般情况下，代码大致应该是这样的（这里，以 1.84 版为例）：

```
Private loACReport, lcConnString, loConn, loRs
Local loACReport, lcConnString, loConn, loRs

* 创建 AC Report 对象实例
m.loACReport = Createobject([AcRptEngineX.AcRptEngine])
m.loACReport.Init

* 定义 ADO 连接字符串
m.lcConnString = [Provider=MSDASQL.1;Persist Security Info=False;Extended
Properties="DSN=MS Access Database;DBQ=E:\ACEngine1.84 For
VFP\Data\ReportDemo.mdb;DefaultDir=E:\ACEngine1.84 For VFP\Data;DriverId=25;FIL=MS
Access;MaxBufferSize=2048;PageTimeout=5;UID=admin;]

* 创建 ADO 连接对象并打开连接
m.loConn = Createobject([ADODB.Connection])
m.loConn.Open(m.lcConnString)

*!* 创建 ADO 记录集
m.loRs=.oConn.execute([select * from Zlemployee])

*!* 加入 ADO 记录集
m.loACReport.AddDataSet([Zlemployee],m.loRs)
```

**!* 如果有已经存在的报表，加入*

```
m.loACReport.SetReportFile(Set("Default") + Curdir() + [Reports\list.apt])
```

```
m.loACReport.Preview          && 预览
```

```
m.loACReport.ShowDesigner    &&设计
```

```
m.loACReport.Print           && 打印
```

OK，到此为止，我们已经可以在 VFP 里使用 AC Report 了。你说什么？ 2.1 及其之后的版本怎么写代码？麻烦往前翻翻看：)

五 还有更好的方式吗？

前面，我们已经知道 AC Report 在 VFP 中使用的基本方式。虽然，我想停下来，但是，我知道这远远不是一个最好的方式，我还需要继续摸索下去.....

对于大多数 Foxer 来说，ADO 连接，是陌生的，况且，我需要连接 SQL Server 数据库，也可能是 MYSQL 数据库，或者其他的关系数据库，针对它们，ADO 的连接字符串该怎么写呢？还有一个问题，我已经通过 ODBC 建立了一个到关系数据库的连接，那么，我再搞出一个 ADO 连接出来，怎么看怎么怪异。但是，如果我不建立 ADO 连接的话，我如何获得 AC Report 所需要的数据源，也就是 ADO 记录集呢？我只有 Cursor.....

万恶的微软，在设计 VFP 的时候，为什么没为 Foxer 想到这个呢？

（芭啦芭啦，这里是你可能的所有牢骚.....）

打住，所有的牢骚全部打住。

如果你够聪明的话，你也许对字符串的问题不会发牢骚。因为前面已经展示了一个方法来获得 ADO 连接字符串，而且，看上去，似乎是一段 VFP 代码把“连接向导”调出来的。是滴，它确实是用 VFP 代码调用出来的，我在 Data

Explorer 的源代码里转悠了将近两个小时，找到了它。简单到你一看到它就觉得自己智商不够.....我个人确实有这个想法：（

```
Private loDataLinks, loADOCoon
Local loDataLinks, loADOCoon

m.loDataLinks = Createobject([DataLinks])
m.loADOCoon = m.loDataLinks.PromptNew()

? m.loADOCoon.ConnectionString
m.loADOCoon.Close()
```

完美！

Cursor 怎么办呢？我们继续.....

微软曾经在 2005 年释放过一个 COM 组件，它的功能之一，就是将 Cursor 和 ADO 记录集相互进行转换，虽然它的目的是为了 VFP 应用和其他语言应用有一个数据交换的方式，但是，在这里，它却是我们更好的使用 AC Report 的一个称手工具。

它的使用也非常的简单：

```
Private loCOMUtil, loRs
Local loCOMUtil, loRs

* 建立 COM 实例
Try
    m.loCOMUtil = Create("vfpcom.comutil")
Catch
    !/N REGSVR32.Exe .\vfpcom\vfpcom.Dll
    m.loCOMUtil = Create("vfpcom.comutil")
Endtry
```

* 建立一个空的 ADO 记录集

```
m.loRs = CreateObject("ADODB.Recordset")
```

* 打开 VFP 表(当然，大多数人希望是 Cursor，但是，VFP 表打开后也是 Cursor 啊。。。啊。。。)


```
Use abc.dbf In 0
```

```
Select abc
```

```
* Cursor To ADO
```

```
m.loCOMUtil.CursorToRS(m.loRs)
```

```
Use
```

一切都是这么的完美，我想，你会迫不及待的想去尝试 AC Report 了。

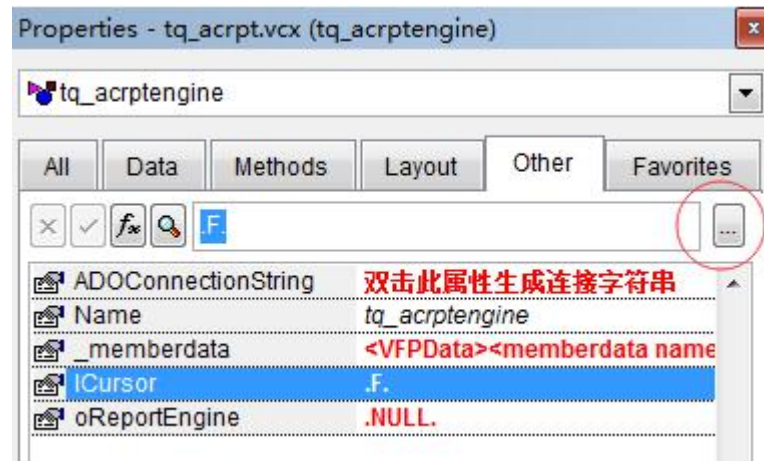
老乡，别跑！

六 TQ_acrpt 类库

前文已经详细讲述了如何使用 AC Report ，但是，作为一个资深的 Foxer —— 资深不代表水平高，意思还可能仅仅是我比你早来到这个世界几分钟 —— 我还是对它进行了二次封装，我喜欢 OOP ，我喜欢 VFP 的 OOP ：)

在 TQ_acrpt 类库中，存在两个类：acrptEngineX184 和 acrptEngineX3212 。它们分别是 AC Report 1.84 版和 3.2.12 版的 VFP 封装。你可以在自由使用它，而不用在乎我的感受：)

需要说明的是，类存在一个自定义属性：ICursor ，默认值是 .F. ，一旦你更改它的值为 .T.，那么，你双击 ADOConnectionString 属性时，就无法调出 ADO “连接向导”了。我不喜欢这种连接方式，但是，我还是为它做了一些东西。



本来到这里，基本应该结束了，但是我在爽了若干个小时，发觉心底有一丝丝的躁动.....有有一些什么东西在作祟.....封装的类库，不够简洁！

于是，在抽了支香烟之后，我决定耍个小伎俩.....删除 acrptEngineX184 类，并重命名 acrptEngineX3212 为 TQ_acrptEngine 。我是如何合二为一的呢？仅仅是一个 Try...Catch...EndTry 的嵌套结构：

```
Private 110K
Local 110K As Logical

m.110K = .T.

Try
    This.oReportEngine = Createobject([AcReport.AcRptEngine])
Catch
    Try
        This.oReportEngine = Createobject([AcRptEngineX.AcRptEngine])
    Catch
        m.110K = .F.
    Endtry
Endtry
```

无论你使用的是 AC Report 的哪个版本，这个类都可以非常好的工作！不要怀疑，事实上我已经验证通过了:)

是该说再见的时候了,Until next time..... VFP Rocks!