

Contents

Visual FoxPro

- 从 TreeView 和 ListView 访问 ImageList
- 基于游标的网格为空
- 将 Visual FoxPro 数据库添加到 SQL Server
- 在 Visual FoxPro 报告中应用格式文本效果
- Service Pack 1 中的 Bug 列表
- C0000005 错误和故障排除
- C0000006 错误和疑难解答建议
- 捕获组合框中的输入
- 转换函数从 SQL Server 获取日期
- 创建主索引
- 在 Visual FoxPro 中创建启动屏幕
- 在程序中创建存储过程
- 确定 Visual FoxPro 的版本
- 导出包含备注字段的记录
- 无法打印报表或编辑窗口中的内容
- 设置日期或时间值以查询表
- 改进应用程序中的性能
- 查找值或最接近的匹配项
- 将参数传递给 SQL Server 存储过程
- 禁止从报表预览打印
- 以编程方式修改报告字段
- 如果日期不是美国日期格式, 查询将失败
- 使用 MScomm32 从串行端口接收
- 刷新的网格不显示任何数据
- 手动注册 ActiveX 控件
- 发送到串行端口
- Visual FoxPro 9.0 的 Service Pack 1
- SQL Server 分布式查询

[传输和接收二进制数据](#)

[解决可疑损坏问题的疑难解答](#)

[对窗体控件使用 KeyPress 事件](#)

[使用 FoxPro 检查是否存在目录](#)

[在 Visual FoxPro 中使用 null 值](#)

[使用 "追加常规" 命令](#)

[使用 DateTimePicker 控件](#)

[使用 "将打印机设置为名称" 命令进行打印](#)

[在 SQL 语句中使用通配符](#)

[更多产品](#)

使用 Visual FoxPro 代码访问 TreeView 和 ListView 控件的 ImageList

2020/10/22 •

原始产品版本: Visual FoxPro

原始 KB 数: 170146

摘要

本文说明如何以编程方式通过 TreeView 和 ListView 控件访问 ImageList 控件的图像。

更多信息

本节包含两个代码示例。第一个示例说明如何将 ImageList 图像分配给 TreeView 控件。在此示例中, 如果将叶子添加到 TreeView 的节点中, 则会将已关闭的文件夹图标分配给节点。如果包含叶子的节点已展开, 则其图标将更改为打开的文件夹, 折叠时, 它将更改回关闭的文件夹。

第二个示例演示如何将 ImageList 图像分配给 ListView 控件。在此示例中, 当创建列表项时, 将为其分配一个大图标。然后, 可以在小图标和大图标之间切换项目图标。对于每种类型的图标--使用大型和小型的 ImageList 控件; 否则, 这两种类型的图标都显示为相同大小。

NOTE

包含图标和位图文件的文件夹在 Visual FoxPro 6.0 中已更改。

图标的新位置为: `home(4)+"icons\office\<filename>"`

位图的新位置为: `home(4)+"bitmaps\outline\<Filename>"`

以下是第一个示例的示例代码, 用于将 ImageList 图像分配给 TreeView:

```
LVForm=CREATEOBJECT("MyLVForm")
LVForm.SHOW
READ EVENT

*** Define the Form to hold and display all the controls ***
DEFINE CLASS MyLVForm AS FORM
LEFT=23
HEIGHT=250
WIDTH=425
mndx=0
*** Add the Control Objects to the Form ***
ADD OBJECT ImageList1 AS MyImageList
ADD OBJECT TreeView1 AS MyTreeView
ADD OBJECT AddNode AS ANCButton
ADD OBJECT AddLeaf AS ALCButton
ADD OBJECT mDelete AS DCButton
ADD OBJECT mExit AS ECButton
ENDDDEFINE

*** Define ImageList to Hold Images for the TreeView Control ***
DEFINE CLASS MyImageList AS OLECONTROL
OLECLASS="COMCTL.ImageListCtrl.1"
TOP=36
LEFT=420
```

```

PROCEDURE INIT
THIS.OBJECT.ListImages.ADD(1,,LOADPICTURE(home()+;
"samples\graphics\bmps\outline\closed.bmp"))
THIS.OBJECT.ListImages.ADD(2,,LOADPICTURE(home()+;
"samples\graphics\bmps\outline\open.bmp"))
ENDPROC
ENDDEFINE

```

```

*** Define the TreeView Control ***
DEFINE CLASS MyTreeView AS OLECONTROL
OLECLASS="COMCTL.TreeCtrl.1"
TOP=24
HEIGHT=169
WIDTH=300
LEFT=20
.OBJECT.LineStyle=1
VISIBLE=.T.

```

```

*** Set ImageList property to ImageList1 Images ***
PROCEDURE INIT
THIS.OBJECT.ImageList=THISFORM.ImageList1.OBJECT
ENDPROC
PROCEDURE nodeclick
LPARAMETERS NODE
ENDPROC

```

```

*** Assign Image for Collapsed object ***
PROCEDURE collapse
LPARAMETERS NODE
IF !EMPTY(NODE.IMAGE)
NODE.IMAGE=1
ENDIF
ENDPROC

```

```

*** Assign Image for Expanded object ***
PROCEDURE expand
LPARAMETERS NODE
IF !EMPTY(NODE.IMAGE)
NODE.IMAGE=2
ENDIF
ENDPROC
ENDDEFINE

```

```

*** Define the Add Node Command Button ***
DEFINE CLASS ANCButton AS COMMANDBUTTON
TOP=70
LEFT=350
HEIGHT=18
WIDTH=59
CAPTION="Add Node"
VISIBLE=.T.
PROCEDURE CLICK
mndxtxt=""
mndxtxt=ALLTRIM(STR(THISFORM.mndx))

```

```

THISFORM.TreeView1.nodes.ADD(,2,mndxtxt+;
" ",REPLICATE(mndxtxt,5),,0)
THISFORM.mndx=THISFORM.mndx+1
THISFORM.TreeView1.ENABLED=.T.
THISFORM.AddLeaf.ENABLED=.T.
THISFORM.mDelete.ENABLED=.T.
ENDPROC
ENDDEFINE

```

```

*** Define the Add Leaf Command Button ***
DEFINE CLASS ALCButton AS COMMANDBUTTON
TOP=100
LEFT=350
HEIGHT=18
WIDTH=59
CAPTION="Add Leaf"

```

```

VISIBLE=.T.
ENABLED=.F.
PROCEDURE CLICK
so=THISFORM.TreeView1.SELECTEDITEM
IF !ISNULL(so)
THISFORM.TreeView1.nodes.ADD(so.INDEX,,
4,, "Sub"+ALLTRIM(so.TEXT),,0)
so.IMAGE=1
ENDIF
THISFORM.REFRESH
ENDPROC
ENDDDEFINE
*** Define the Delete Command Button ***
DEFINE CLASS DCButton AS COMMANDBUTTON
TOP=150
LEFT=350
HEIGHT=18
WIDTH=59
CAPTION="Delete"
VISIBLE=.T.
ENABLED=.F.
PROCEDURE CLICK
o=THISFORM.TreeView1.SELECTEDITEM
IF !ISNULL(o)
IF !ISNULL(o.PARENT)
IF o.PARENT.children<2
o.PARENT.IMAGE=0
ENDIF
ENDIF
THISFORM.TreeView1.nodes.REMOVE(o.INDEX)
ENDIF
ENDPROC
ENDDDEFINE
*** Define the Exit Command Button ***
DEFINE CLASS ECButton AS COMMANDBUTTON
TOP=215
LEFT=180
WIDTH=70
HEIGHT=25
CAPTION="Exit"
VISIBLE=.T.
PROCEDURE CLICK
THISFORM.RELEASE
CLEAR EVENTS
ENDPROC
ENDDDEFINE

```

下面是第二个示例的示例代码：

```

LVForm=CREATEOBJECT("MYLVForm")
LVForm.SHOW
READ EVENT

*** Define the Form that will hold and display the controls ***
DEFINE CLASS MyLVForm AS FORM
LEFT=23
HEIGHT=300
WIDTH=478
lvitm=0
lvitxt=''
ADD OBJECT ImageList2 AS MyImageList1
ADD OBJECT ImageList3 AS MyImageList2
ADD OBJECT ListView1 AS MyListView
ADD OBJECT ADDITEM AS AICButton
ADD OBJECT LSIOption AS LSIOGroup
ADD OBJECT mDelete AS DButton
ADD OBJECT MExit AS ExButton

```

```

ENDDDEFINE

*** Need separate ImageList Controls for Large and Small Icons of
*** ListView otherwise both type of icons will appear as same size
***
*** Define ImageList to Hold Large Icon Image for the ListView
*** Control

DEFINE CLASS MyImageList1 AS OLECONTROL
OLECLASS="COMCTL.ImageListCtrl.1"
TOP=204
LEFT=420
PROCEDURE INIT
THIS.OBJECT.ListImages.ADD(1,,LOADPICTURE(home(4)+;
"icons\office\folder01.ico"))
ENDPROC
ENDDDEFINE

*** Define ImageList to Hold Small Icon Image for the
*** ListView Control

DEFINE CLASS MyImageList2 AS OLECONTROL
OLECLASS="COMCTL.ImageListCtrl.1"
TOP=264
LEFT=420
PROCEDURE INIT
THIS.OBJECT.ListImages.ADD(1,,LOADPICTURE(home(4)+;
"bitmaps\outline\closed.bmp"))
ENDPROC
ENDDDEFINE

*** Define the ListView Class ***
DEFINE CLASS MyListView AS OLECONTROL
OLECLASS="COMCTL.ListViewCtrl.1"
TOP=24
HEIGHT=169
WIDTH=275
LEFT=36
.OBJECT.LineStyle=1
VISIBLE=.T.

*** Set the ListView Icons property to Images in ImageList1 ***
*** and SmallIcons property to Images in ImageList2 ***

PROCEDURE INIT
THIS.OBJECT.icons=THISFORM.ImageList2.OBJECT
THIS.OBJECT.smallicons=THISFORM.ImageList3.OBJECT
ENDPROC
ENDDDEFINE

*** Define the Add Item Button Class ***
DEFINE CLASS AICButton AS COMMANDBUTTON
TOP=50
LEFT=360
HEIGHT=18
WIDTH=59
CAPTION="Add Item"
VISIBLE=.T.
PROCEDURE CLICK
THISFORM.lvitm=THISFORM.lvitm+1
THISFORM.lvitxt=ALLTRIM(STR(THISFORM.lvitm))
x=THISFORM.ListView1.OBJECT.listitems.ADD;
(,THISFORM.lvitxt+"_", "Item"+THISFORM.lvitxt,)
x.smallicon=1
x.ICON=1
THISFORM.LSIOption.ENABLED=.T.
THISFORM.mDelete.ENABLED=.T.
THISFORM.REFRESH
ENDPROC

```

```

ENDPROC
ENDDDEFINE

*** Define the Large/Small Icon Option Group ***
DEFINE CLASS LSIGroup AS OPTIONGROUP
BUTTONCOUNT = 2
VALUE = 1
HEIGHT = 39
LEFT = 340
TOP = 80
WIDTH = 96
Option1.CAPTION = "Large Icon"
Option1.LEFT = 11
Option1.TOP = 5
Option1.WIDTH = 82
Option1.NAME = "Option1"
Option2.CAPTION = "Small Icon"
Option2.LEFT = 11
Option2.TOP = 20
Option2.WIDTH = 82
Option2.NAME = "Option2"
PROCEDURE INTERACTIVECHANGE
DO CASE
CASE THIS.VALUE=1
THISFORM.ListView1.OBJECT.VIEW=0
CASE THIS.VALUE=2
THISFORM.ListView1.OBJECT.VIEW=1
ENDCASE
THISFORM.REFRESH
ENDPROC
ENDDDEFINE

*** Define the Delete Button Class ***
DEFINE CLASS DButton AS COMMANDBUTTON
TOP=156
LEFT=360
HEIGHT=18
WIDTH=59
CAPTION="Delete"
VISIBLE=.T.
ENABLED=.F.
PROCEDURE CLICK
o=THISFORM.ListView1.SELECTEDITEM
IF !ISNULL(o)
THISFORM.ListView1.listitems.REMOVE(o.INDEX)
ENDIF
ENDPROC
ENDDDEFINE

*** Define the Exit Button Class
DEFINE CLASS ExButton AS COMMANDBUTTON
TOP=226
LEFT=192
WIDTH=192
HEIGHT=18
WIDTH=59
CAPTION="Exit"
VISIBLE=.T.
PROCEDURE CLICK
THISFORM.RELEASE
CLEAR EVENTS
ENDPROC
ENDDDEFINE

```

如果选择-SQL 命令在 Visual FoxPro 中重置光标，则基于游标的网格是空的。

2020/10/22 •

如果选择-SQL 命令在 Visual FoxPro 中重置了光标，本文将帮助您解决基于游标的网格为空的问题。

原始产品版本： Visual FoxPro

原始 KB 数： 140653

症状

在 Microsoft Visual FoxPro 中，如果 `RecordSource` 将网格的属性设置为游标，然后选择-SQL 命令重置光标，如果 `RecordSource` 属性未设置回其自身，则网格显示为空白。

如果运行 SELECT-SQL 以重建游标，下面的命令将正确刷新网格，但如果自定义了网格的列或标题属性，则这些自定义更改将丢失，并且网格的 `column` 和 `header` 属性将更改回其默认设置：

```
THISFORM.GRID1.RECORDSOURCE=THISFORM.GRID1.RECORDSOURCE
```

此类自定义可以是标头不同的标头、列长度较长的标头以及列或标题事件中的代码。有关详细信息，请参阅 Microsoft 知识库中的以下文章：[131836 PRB: 不刷新显示查询中的游标的网格](#)

原因

当您使用 SELECT-SQL 命令重新生成网格时，必须先销毁原始游标，然后才能创建新游标。发生这种情况时，也会清除网格列和标题，然后再重新创建它们。即使将 `RecordSource` 网格的属性重新设置为其自身，网格的列和标题的自定义设置也会丢失。

解决方案

若要刷新网格，而不丢失列和标题的自定义属性，请将 `RecordSource` 属性设置为已创建的虚拟游标，该游标已创建时使用的字段与网格所基于的游标相同。在选择-SQL 运行后，将属性更改 `RecordSource` 回重建的游标。这样，在重新生成游标时，列和标题的属性就可以保持不变。

状态

此行为是设计使然。

有关重现行为的步骤的详细信息

1. 打开位于 `\Vfp\Samples\Data` 目录中的 "Customer" 表，创建名为 "GridForm" 的窗体。
2. 将以下代码放在窗体的 Load 事件中：

```
SELECT cust_id, city, country FROM customer INTO CURSOR Temp1  
SELECT cust_id, city, country FROM customer WHERE country = "" ;  
INTO CURSOR Temp2
```

3. 将以下代码放在表单的销毁事件中：


```
SELECT Temp1
USE
SELECT Temp2
USE
```

4. 将网格放置在窗体上，并为网格指定以下属性设置：

```
ColumnCount=3
RecordSourceType=Alias
RecordSource=Temp1
```

5. 更改 grid1 的标题。Column1 header1 to Customer Id, 并将 Column1 Width 属性设为100。

6. 向窗体中添加一个文本框。

7. 添加一个命令按钮，并将以下代码放在其 Click 事件中：

```
THISFORM.GRID1.RECORDSOURCE = "Temp2"
SELECT cust_id, city, country FROM customer ;
WHERE country = Thisform.text1.value ;
INTO CURSOR Temp1
THISFORM.GRID1.RECORDSOURCE = "Temp1"
```

8. 运行窗体。在文本框中键入 "西班牙"，然后单击 "命令" 按钮。网格应显示 "国家/地区" 字段等于 "西班牙" 的所有记录。列1的标题名称保持不变，并且该列的宽度不变。通过将网格的记录源更改为 "Temp2"，在重建 Temp1 游标时，不会重置列和标题的属性。

9. 将 `THISFORM.GRID1.RECORDSOURCE` 命令按钮的 Click 事件中的两行更改为注释，并将以下代码添加为代码的最后一行：

```
THISFORM.GRID1.RECORDSOURCE = THISFORM.GRID1.RECORDSOURCE
```

在运行窗体后，在文本框中键入国家/地区名称，然后单击命令按钮，请注意，列名称发生变化，宽度较小。

将 Visual FoxPro 数据库添加到 SQL Server 作为链接服务器

2020/10/22 •

本文介绍了如何以编程方式在 Visual FoxPro 中添加和查询 Visual FoxPro 数据源作为链接服务器。

原始产品版本: Visual FoxPro

原始 KB 数: 199131

摘要

SQL Server 7.0 和 SQL Server 2000 允许将外部数据源添加为链接服务器。此功能提供对 OLE DB 数据源的分发、异类查询的访问。本文介绍如何以编程方式在 Visual FoxPro 中添加和查询 Visual FoxPro 数据源作为链接服务器。

更多信息

`sp_addlinkedserver` 是 SQL Server 7.0 和 SQL Server 2000 中引入的新存储过程。`sp_addlinkedserver` 创建链接服务器, 允许对 OLE DB 数据源进行分发和异类查询。

从 Transact-sql 中添加链接服务器的语法为:

```
sp_addlinkedserver [@server =] 'server',
  [@srvproduct =] 'product_name',
  [@provider =] 'provider_name',
  [@datasrc =] 'data_source',
  [@location =] 'location',
  [@provstr =] 'provider_string',
  [@catalog =] 'catalog'[@server =] 'server' Is the name of the linked server to
create with sp_addlinkedserver.

[@srvproduct =] 'product_name' Is the product name of the OLE DB data
source to add as a linked server.

[@provider =] 'provider_name' Is the unique provider identifier of the
OLE DB provider corresponding to the
data source.

[@datasrc =] 'data_source' Is the name of the data source, as
interpreted by the OLE DB provider.

[@location =] 'location' Is the location of or path to the
database as interpreted by the OLE DB
provider.

[@provstr =] 'provider_string' Is the OLE DB provider-specific.

[@catalog =] 'catalog' Is the catalog to be used when making a
connection to the OLE DB provider.
```

下面的代码段将 Visual FoxPro 数据库 `Testdata.DBC` 从 SAMPLES\DATA 目录添加到 SQL server 作为链接服务器。此代码段假定 Visual FoxPro 和 SQL Server 正在同一台计算机上运行。

NOTE

用户名和 ■ 必须具有对数据库执行这些操作的权限。

```
Source_Path=IIF(VAL(SUBSTR(VERSION(),15,2))=6,HOME(2),HOME()+"SAMPLES\")
Connect_String='DRIVER={SQL Server};' + ;
'SERVER=MY_SERVER;DATABASE=PUBS;UID=UserName;PWD=StrongPassword'
gnConnHandle=SQLSTRINGCONN(Connect_String)
IF gnConnHandle > 0
* Create a command string to pass to SQL Server via SQLExec
SQLCommand="sp_addlinkedserver 'VFP','', 'MSDASQL',NULL,NULL," + ;
"'DRIVER={Microsoft Visual FoxPro Driver};" + ;
"SourceDB="+Source_Path+"DATA\TESTDATA.DBC;SourceType=DBC;NULL'"* CREATE the LINKED Server"
Create_Linked_Server=SQLExec(gnConnHandle,SQLCommand)
IF Create_Linked_Server > 0
* The linked server was successfully created
* Run the query
=RunQuery()
ELSE
* The Linked Server either already exists or the command failed.
* Test for existence of linked server with aerror()
=AERROR(s_failed)
IF "VFP" ALREADY EXISTS."$UPPER(s_failed[1,2])
* The linked server exists, so run the query
=RunQuery()
ELSE
* The linked server doesn't exist, so display a message
=MESSAGEBOX(s_failed[1,2],32,'Failed')
ENDIF
ENDIF
=SQLDISCONN(gnConnHandle)
ENDIF

PROCEDURE RunQuery
SQLCommand="SELECT * FROM OPENQUERY(VFP,'SELECT * FROM CUSTOMER')"
QRYVal=SQLExec(gnConnHandle,SQLCommand,'SQLRESULTS')
IF QRYVal > 0
SELECT SQLResults
BROW
ELSE
=AERROR(L_Server)
=MESSAGEBOX(L_Server[1,2],32,'Query Failed')
ENDIF
RETURN
```

运行代码段后，打开 SQL Server 企业管理器并展开 "链接服务器" 节点。已添加名为 "VFP" 的链接服务器。

参考

有关的详细信息 `sp_addlinkedserver`，请 `sp_addlinkedserver` 在 Transact-sql 参考帮助文件中搜索。

在 Visual FoxPro 报告中应用格式文本效果

2020/10/22 •

本文介绍如何为包含 rtf 格式的表格中的字段的各个部分加下划线，并在报表上显示该格式。

原始产品版本： Visual FoxPro

原始 KB 数： 246088

摘要

随时，最好能够向 Visual FoxPro 报告中的单个单词添加格式效果。可以使用格式文本 ActiveX 控件、常规字段和 Visual FoxPro 报告上的 OLE 绑定控件来实现这一点。

更多信息

本示例演示如何为表中字段的各个部分加下划线，并在报表上显示该格式。您可以将格式文本中可能存在的任何效果或效果组合应用于您的示例。

1. 打开 Visual FoxPro 并创建一个新报表。
2. 向报表的详细信息区段中添加 图片/Active-X 绑定控件。将其 FIELD 属性设置为 Dcolor。
3. 将报告另存为 *rtfDemo*。
4. 将以下代码粘贴到新程序中。将该程序保存在与报告相同的目录中，然后运行它。

```

CLOSE ALL
CLEAR ALL
tSafety = SET("safety") &&Store SET status of Safety to a variable
SET SAFETY OFF &&Check to see if the Dcolor table exists
IF FILE("dcolor.dbf")
USE dcolor EXCLUSIVE
ELSE
CREATE TABLE dcolor (dcolor c(20),gcolor g) &&Create and add records if DColor table doesn't exist
INSERT INTO dcolor (dcolor) VALUES ("Red")
INSERT INTO dcolor (dcolor) VALUES ("Green")
INSERT INTO dcolor (dcolor) VALUES ("Blue")
INSERT INTO dcolor (dcolor) VALUES ("Purple")
INSERT INTO dcolor (dcolor) VALUES ("Orange")
INSERT INTO dcolor (dcolor) VALUES ("Yellow")
ENDIF
_rtffile = SYS(3)+".rtf" &&Create a legal file name with a .RTF extension
*NOTE: The file must have a .RTF extension to work properly
SCAN &&Create a Richtext OLE bound control in the general field of the dColor table
SET TEXTMERGE TO &_rtffile NOSHOW
SET TEXTMERGE ON
*!!IMPORTANT!! The following richtext formatting line(s) need to be
*included as one line in your program.
\\{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl
{\f0\fnil\fcharset0 Times New Roman; }} \viewkind4\uc1\pard\f0\fs20
The big \ul <<ALLT(dcolor.dcolor)>>\ulnone dog\par}
SET TEXTMERGE TO
APPEND GENERAL gcolor FROM &_rtffile CLASS "RICHTEXT.RICHTEXTCTRL.1"
ENDSCAN
SET SAFETY &tSafety
ERASE &_rtffile

REPORT FORM rtfdemo PREVIEW NOCONSOLE
USE
CLOSE ALL

```

打印报表时，文本中的颜色会加下划线。

NOTE

在某些情况下，RTF 格式文本的打印预览看起来不正确。但是，打印字符串时，输出的格式正确。

TIP

若要生成与上面的代码中的字符串类似的字符串，请为写字板中的表达式创建模板 (或其他能够创建 Rtf 文件) 的应用程序。为您的文本添加下划线、颜色并应用任何其他格式，然后将其保存到 *.RTF* 文件。打开 *.RTF* 文件中的 rtf 文件，并将富文本代码复制/粘贴到您的 Visual FoxPro 程序中。

(此应用程序需要 RichTX32 控件。Visual FoxPro 3.0 和 3.0 b 不附带 RichTX32。)

Microsoft Visual FoxPro 9.0 Service Pack 1 中修复的 bug 列表

2020/10/22 •

本文提供有关在 Visual FoxPro 9.0 Service Pack 1 中修复的版本和错误的详细信息。

原始产品版本: Visual FoxPro

原始 KB 数: 907737

摘要

本文包含有关在 Microsoft Visual FoxPro 9.0 Service Pack 1 (SP1) 中修复的 bug 的信息。有关详细信息, 请参阅 [how to 获取 Service Pack 1 For Visual FoxPro 9.0](#)。

Visual FoxPro (VFP) 9.0 SP1 包含针对以下问题的修补程序。

控件和类

- 无法使用鼠标将焦点从 DHTML 控件中的其他控件获得焦点。
- 如果基础表中包含 (s) 的 AutoInc 字段, 则向导生成的表单将在尝试添加记录时失败。
- Web 浏览器控件: 实现 DWebBrowserEvents2 会导致参数计数不匹配。
- 将表单保存到使用中的文件会导致最终发生严重异常。
- 生成包含仅由回车符分隔的属性的 VCX 的项目时, 将发生严重异常。
- 集合: 从集合中删除项目 ≥ 500 项目在应用程序验证工具下崩溃 VFP。
- CursorAdapter: TABLEUPDATE (0) 忽略表缓冲游标的 SendUpdate 设置。
- TextBox: C0000005 exception with "自动完成" TextBox, TextBox 将 "控件来源" 设置为允许空值的字段。
- ComboBox: 当 ControlSource 为 null 成员属性时, 在下拉框中设置值时出现 "数据类型不匹配" 错误消息。
- 当 MSXML3.dll 文件未注册时, MODI 格式命令发生致命异常错误。
- 向导生成的表单中的搜索功能适用于新的 VFP9 数据类型。搜索不会找到 varchar 数据。
- 控件类不会调整其自身内的定位控件的大小。
- 在销毁表单时, QUIT 不会关闭 VFP 或应用程序。
- 已转换的表单的读取在 VFP 9.0 中发布, 而不是在 VFP 的早期版本中发布。
- 当选中的别名从一个填充网格更改为同一表的别名时, 网格将丢失数据, 并对第二个别名使用计算命令。
- C0000005 异常在 ListBox ListIndex = 0 时按 ENTER 键。
- 值为 256/512 和奇数宽度/高度的意外定位行为。
- 表单: 内部一致性错误, 然后在使用 aselobj (# A1 引用添加与表单的环境关系后编辑表单时出现致命异常。
- 使用 Word 实现失败。使用 Office 2003 的 Application 对象。
- 将父窗体的自定义方法设置为网格列的 "控件来源" 时, 网格将变为空。
- ActiveX 控件: LeadTools ActiveX 控件在 "窗口" 菜单中显示两个空条形图。
- 复选框: 图形复选框和选项按钮在启用主题时不显示为较暗的背景色。
- 使用存储在隐藏文件夹中的类库的 "内部一致性" 错误消息。
- ComboBox: 除非用户第一次传递 "鼠标悬停于" 下拉列表中的内容, 否则不会触发 DownClick () 。
- Pageframe with TabStretch = 0 (多个), 并且多个选项卡在窗体打开以供编辑时保持增长。
- 为 ADO 数据源添加对 adVarNumeric 的支持。
- 在窗体集中的 PageFrame 上刷新问题将导致 TextBox 重影。
- 当 .exe 通过网络运行时, 在窗体上的不同控件上刷新问题 (UNC 或映射驱动器) 。

- 下拉列表样式 ComboBox 的值不会显示在可访问的事件查看器中。
- 孤立的表单环境导致 VFP 在退出时崩溃。
- 在 pageframe 上禁用 Ctrl + Tab off 页时出现 "内部一致性" 错误消息。
- 一个窗体窗体的对象会在第二次运行时清除用户属性的内存损坏。
- 在修改或运行包含无效 Zorder 设置的对象的表单时出现 "内部一致性" 错误消息。
- 工具栏在顶级窗体中不处于活动状态的工具栏。
- 网格: "数据类型不匹配" 错误消息9通过网格或浏览编辑双字段。
- 包含模式窗体的顶级窗体应可以调整大小并可移动。
- "内部一致性" 错误消息 "在网格中初始化控件" 列中的 "错误消息设置值" 属性。
- "内部一致性" 错误消息: 单击复合索引键上的 "数据环境中的关系" 窗口中的 "持久关系"。
- CursorAdapter: 在自动生成的 SQL 中, 基表名称转换为小写形式。
- 如果表单使用专用 datasession, 则绑定自动完成 TextBox 中的 "内部一致性" 错误消息。
- EditBox wordwrap 导致空行消失。

Engine

- 数据引擎不再能优化涉及已删除 (# A1 函数和 OR 子句的一些查询。
- STR (# A1 上的索引未 Rushmore 优化为 SET ANSI OFF。
- 使用 UNION 时, 子查询中的前 N 将被忽略。
- 在具有不同代码页的表之间进行复制时不转换备注值 (SQL SELECT, 从) 中追加。
- 如果存在行验证规则, 则替换为未定义的变量的致命异常。
- "挂起" 在执行 "解锁" 时发生。
- "内部一致性" 错误消息: 如果表是在不同的数据会话中进行 transactable, 则在事务中访问 free 表。
- 在准备好的远程视图上重新查询 (# A1 失败。
- 当使用货币的索引进行优化时, 找到的匹配项将失败, 而比较的右侧部分则是特定的数值。
- 使用命令可以为工作区分配重复的别名。
- END 事务上的 "表具有文件长度/记录计数不一致" 的错误消息。
- 对大型表创建 SQL 视图时出现意外的 "SQL expression 太复杂" 错误消息。
- 在使用 SET TABLEVALIDATE >= 8 的 ZAP 后重新打开或刷新表将导致 "表。。已损坏" 错误消息。
- 当关联子查询返回 COUNT (*) 与 0 进行比较时, SQL SELECT 将返回不正确的结果。
- 在事务中更新绑定到网格的 CursorAdapter 时的数据损坏。
- 意外的 "SQL: 错误关联域" 错误消息, 其中包含两个 EXISTS 子查询。
- 未验证关联, 无法满足 "仅支持对直接父查询的关联" 限制。
- 当聚合用于 HAVING 但不在选择列表中时, 意外的 "子查询返回了多条记录" 错误消息。
- 如果将 ORDER BY + TOP N 计算与 GROUP BY 求值合并, 并对 TOP 进行优化, 则查询将返回不正确的结果。
- SELECT COUNT (ICASE (, 1) # A3 返回不正确的结果。
- 如果在 FROM 中使用子查询, 则选择当前工作区中的数组更改。
- 在 REPLACE with scope 之后表缓冲游标上的索引已损坏。
- 带有外部联接的查询生成不正确的结果如果不为空 (...) 在 WHERE 中使用。
- 使用相关子查询的错误消息中, 意外的 "SQL expression 太复杂"。
- "检测到 Microsoft Visual c + + 运行时库缓冲区溢出!" 并 C0000005 在对大型表运行扫描循环时出现异常。
- 转换 (空白 datetime 字段的日期为 # A1 返回 datetime 类型。
- 更改表。。更改列以将字符字段更改为整数也会更改值。
- "为索引生成键时出错 ..." 在使用 str 编制索引时出现错误消息, 在 NULL 值上 (# A1)。
- Tableupdate 中的 "内部一致性" 错误消息。在将 (# A3 后重新查询 (# A1 到具有 batchupdatecount = -1 和表缓冲的远程视图。
- SQL 中的 "内部一致性" 错误消息选择 "记录大小 * 记录数为高" 时。

- 在 ENGINEBEHAVIOR < 90 时, 为具有非当前代码页的表启用优化。
- 使用嵌套的集合筛选器编制索引时出现 "内部一致性" 错误消息。
- 临时更改当前工作区的筛选条件可能会中断关系, 也可能中断嵌套的 XML 生成与 XMLAdapter。
- 查询处理器在选择列表中找不到列(如果它括在括号中), 报告 false "SQL: ORDER BY 子句无效" 错误消息。
- 意外的 "SQL: GROUP BY 子句丢失或无效" 尝试使用 NODATA 创建视图或使用视图的错误消息。
- 查询在 ENGINEBEHAVIOR < 90, 子查询返回聚合函数, 并且没有记录匹配子查询 WHERE 条件时, 会产生不正确的结果。
- 复制到。。由于 nCodepage 无法使用正确的代码页对备忘录进行编码, 而是使用 CPCURRENT (# A1。
- 在使用 "当前代码" 页从表复制到具有不同代码页的表时, 不会转换备注值, (从) 中复制到、SQL SELECT、APPEND。
- 选择。。WITH 缓冲意外从游标中提取缓冲数据, 而 FROM 子句中未显式引用这些数据。

IDE-R

- 项目经理:生成使用仍在内存中的图标的应用程序的致命异常。
- "属性" 窗口:致命异常关闭缩放窗口, 并在应用程序验证器中使用 Enter 键, 并将 KEYCOMP 设置为 DOS。
- 在发出清除弹出窗口时出现 "菜单管理器内部一致性错误" 故障。
- 提示保存查询引用 tmp 文件。
- 当您在 "多显示器安装" 中的 "左监视器" 中运行 VFP 时, 工具提示将显示在右侧监视器中, 其中主监视器位于右侧。
- 具有长表达式的视图设计器中的 C0000005 错误。
- 项目经理:在停靠/取消停靠操作后, 在项目管理器上更改字体的致命异常。
- 项目经理:当为 "选项" 对话框的 "IDE " 选项卡上的 "项目类型" 选择 "覆盖单个设置" 时, 新项目经理不记得它们的位置。
- RI 生成器使用不正确的 "解锁" 命令语法。
- Wizbtns 的 Picbtns 类在与 CursorAdapter 一起使用时显示警告。
- 项目经理: VFP 在使用 classlib 后重建应用程序时, 停止响应(挂起)。
- 在程序编辑器中激活智能感知时出现访问冲突 C0000005。
- IntelliSense:在尝试访问 LPARAMETER 的成员方法的智能感知方法失败后, C0000005 访问冲突退出类设计器。
- 调试大型代码文件时出现致命异常。
- 构建者: MemberData 编辑器-GetHierarchy 中的代码 bug。在 MemberDataEditor 中, (# A1 导致 MemberData 编辑器获取层次结构的错误。
- "属性" 窗口:在值长度超过255个字符的属性上按 ESC 后, C0000005 异常关闭类设计器(CursorAdapter、XMLAdapter、XMLTable 和 XMLField)。
- 编辑器:使用仅包含两条空行的 "代码编辑器" 窗口中的 "表达式生成器" 时, 100% CPU 使用率挂起。
- IntelliSense:创建早期绑定 mapi 后, 列表成员中出现致命异常。CreateObjectex 的 session 对象, 其 (# A1。
- 在表浏览过程中消息使用资源文件时, 如果出现未处理的异常, 则出现 "浏览表被关闭" 错误消息 (c05)。
- 在简体中文平台上, 减少包含 dbcs 字符串的 TextBox 的宽度将导致 IDE 停止响应。
- 在简体中文平台上, 执行程序时出现致命异常。
- 调整项目管理器的大小不会更新文件的 treeview。
- "解决方案示例中的选项按钮示例" 引发数据类型不匹配错误。
- 组件库:从上下文菜单中选择 "清理" 键时的 "为索引构建键时出错"。
- 构建者: ReportBuilder 中的 属性 对话框存在问题。
- 在 Citrix 客户端会话中向 Metaframe 4.0 演示文稿服务器打印到 EMF-Citrix 打印机驱动程序时, "打印机未准备就绪" 错误消息。其他应用程序可以打印到这台打印机。
- 视图设计器中的 "查看脚本中不允许使用 '本地 thisview ' 或其位于错误的位置" 错误消息。
- 使 ReportBuilder 可本地化。

语言

- SQLEXEC (# A1 函数允许 reentrance 用于相同的连接句柄。
- DEBUGOUT 将二进制数据输出为字符串。
- CursorToXML (# A1 和 XMLAdapter 不输出以秒为单位的 xsd: dateTime 值(如果设置的秒数为 OFF)。
- C0000005 在 程序编辑器 菜单上执行 "<程序" 中的 "do>" 时, 后跟 "pushjmp/popjmp Call 不匹配" 的错误消息, 然后再调用 EXESCRIP (# A1)。
- WEXIST (# A1 将找到 ToolTipText windows。
- 将 ADO RecordSet 对象的 ActiveConnection 属性设置为 NULL 时出现内存泄漏。
- "内部一致性" 错误消息, 后跟对使用 GetObject ("IIS://LocalHost/W3SVC/1") 返回的对象的 VFP 调用方法的崩溃。
- XMLToCursor (# A1 和 XMLAdapter 如果时间部分存在, 则无法加载 xsd: date 类型的值。
- 如果名称中至少有一个部分包含空格或其他 "非法" 字符, 则 SQLCOLUMNS (# A1 将出现复杂名称。
- ADIR (虚拟电脑) 中返回 >2gig 文件的无效文件大小。
- GETOBJECT (# A1 的参数限制基于文件规范 (261 个字符) 。
- 递归的字符串优化产生错误的结果。
- 文字编号导致 BINTOC (# A1 错误。
- 使用 CTOBIN (# A1 将 Bintoc 转换为字符会引发垃圾邮件值。

VFP OLE DB 提供程序

- OLE DB Provider: SYS (3050) 在提供程序中默认为 128 MB。
- OLE DB Provider: 明显的内存泄漏导致连接错误 "无法转换数据值 ..."对 SQL 链接服务器重复使用 openrowset 或 openquery。
- 无法使用 VFP OLE DB Provider 创建数据集。
- GetSchema (string, string []) 无法根据 INDEX_NAME 进行筛选。
- OLE DB Provider: Enable provider 中的 "FOR and WHERE (3055) (" FOR and WHERE 从句的复杂性 ") 。
- VFP OLE DB Provider 错误地处理二进制参数。
- 启用删除视图和删除过程命令。
- DBSCHEMA_PROCEDURES 行集返回过时的过程。
- DBSCHEMA_PROCEDURES 行集返回不完整的过程定义。
- APPEND 过程将忽略过程中的编译错误。
- 访问不包含任何表的数据库容器 (DBC) 可能导致缓冲区溢出错误。

Reporting

- "报表生成器" 对话框应遵循区域设置。
- 在某些情况下, 报告预览具有不正确的标题。
- 在预览 "对象辅助报表" 时, "初始打印预览" 工具栏与从 "报表预览" 表面菜单中选择 "工具栏" 的打开方式不同。
- 设置为 "缩放内容, 保留形状" 的报表上的对象在报表预览和由 HTMLListener 进行的 HTML 之间以不同的方式显示。
- 打开 "在新页面上启动" 时, 多个详细信息带无法正常工作。
- "转到页面" 对话框在预览容器 TopForm 属性为 .T 时, 将在可见区域中打开。和 windowstate 是正常的。
- 在不带 NOWAIT 子句的 "报表窗体" 命令中预览报表时, 无法访问 "新建样式的报表预览" 工具栏。
- 当您尝试关闭报表设计器并在 "是否将更改保存 <proc> 到" 中单击 "否" 时填充了无用字符的代码窗口? 行。
- 报表的数据分组限制在文档主题 "如何: 向报表中添加数据组" 主题中列为74, 但 ReportBuilder 限制为20。
- 数字格式的对齐方式为 左对齐。这与以前的 VFP 版本和默认报表设计器不同。
- 在将报表变量的初始值设置为无效值之后, 在关闭报表属性对话框时未提供任何错误。

- 包含错误的字段的 "报表表达式" 对话框在 REPORTBEHAVIOR = 90 时不会在预览错误后显示。
- 当重印详细信息标题选项处于打开时, 报告详细信息头中的计算值意外重置。
- ReportListener、AdjustObjectSize 和 MaxHeightAvailable 属性不能与浮动对象一起使用。
- 预览包含位于磁盘上的图像的报表会导致在每次刷新预览时发生 GDI 句柄泄漏。
- 报告预览 GDI 处理对许多页面的限制。
- 关联的页眉和页脚报告选项:连续编号不在页面之间打印。
- ReportListener OutputPage to HDC 使用 96 DPI 而不是打印机 DPI。
- "ResourceManager:: SaveWindowState (# A1" 的第325行中不存在 "属性 WindowState" 错误消息。
- 多选对话框允许设置保护标志, 即使报告受到保护也是如此。
- 如果从 "从 .FRX 加载" 切换到 "链接到类", 则系统会提示您输入 .FRX。
- 取消第一个 GETFILE (.FRX) 提示符后, 再次提示您 GETFILE。
- 如果以前在预览之外声明了 oForm, 则报告预览会覆盖它。
- 程序应在运行时(而不是设计时 96 DPI)考虑 "真实" 屏幕 DPI。
- 从具有特定属性的类加载一个环境时, 会发生各种错误。
- 将链接的 DE 类库的完整路径写入链接代码。这并不总是合适。
- 在某些情况下, 预览表单的属性不会正确保存到资源文件中。
- 如果您选择了 "AllowPrintFromPreview =. F", 则会收到 "未知 member.cmdPrint" 错误消息。
- 并非 FRXOUTPUT 中的所有变量都有 "m"。
- 如果传递了有效的侦听器引用, 则增强应用程序, 使其更好地了解未知输出类型。
- 确保 "m"。在包含表达式的所有方法和 #DEFINEs 中。
- 从使用其 ApplyUserXSLT 属性的 XmlListener 派生的 Filebased 侦听程序无法为 TargetFileName 指定仅目录。
- 在显示 "转到页面" 对话框后, Shift-F10 键无法再显示 "报告预览" 菜单。
- 拉伸的浮动对象的报告带区大小不会正确更改。
- C0000005 在 .frx 文件中具有损坏的 EXPR 字段的报表打印预览过程中出现异常。
- 如果传递了有效的侦听器引用, 则增强 reportoutput 以处理未知输出类型。

运行时和 COM

- 在 Windows 98 第二版上运行的应用程序中不可见的工具栏或状态栏。
- "过程入口点 GetLongPathNameA 无法在 Windows NT 4.0 的运行时中的" 动态链接 KERNEL32.dll 库 "中找到错误消息。
- 在 COM 服务器中实现的内存泄漏。
- 使用无符号的长整型和无符号 int 类型传递无效的 COM 参数。
- 当 COM + 回收 dllhost 工作进程时, REMClearConnect 和 csCriticalSectionEnter 中出现死锁。
- COMARRAY 在 Visual Basic 6.0 COM 服务器上, 日期类型不起作用。
- 无法初始化早期绑定对象的 OCXAPI。
- 传递大型数组 >65000 时, "数组维数无效" 错误消息230。

C0000005 错误以及如何对其进行故障排除

2020/10/22 •

本文将介绍 C0000005 错误并提供故障排除建议。

原始产品版本: Visual FoxPro

原始 KB 数: 2722492

症状

您正在运行 Visual FoxPro 交互开发环境或自定义 VFP 可执行文件。应用程序将关闭并报告此错误:

C0000005 致命异常

原因

C0000005 错误是内存错误。具体来说, C0000005 错误是由缓冲区溢出导致的访问冲突错误。

解决方案

许多情况下可能会导致 Visual FoxPro 报告 C0000005 错误。确定根本原因的最有效方法是在错误发生时, 在 Visual FoxPro 进程中获取内存转储, 然后检查错误模块的堆栈。例如, 如果打印机驱动程序模块位于堆栈顶部, 则应尝试更新打印机驱动程序或尝试使用其他驱动程序。如果第三方 OCX 显示为错误模块, 则应与该控件的开发人员核实, 以获取与您的 Visual FoxPro 版本兼容性的更新。

由于 Visual FoxPro 是32位应用程序, 因此必须使用32位调试器来获取内存转储。

若要获取最新版本的调试诊断, 请参阅 [调试诊断工具 v1.1 1.2 现已推出](#)

以下是有关获取可视化 FoxPro 进程的内存转储的一些一般性说明。

将其安装在 (s) 遇到故障的计算机上。双击 DebugDiag.msi 并按照安装向导操作。将 DebugDiag 安装到至少 200 megs 免费的硬盘驱动器。

配置 DebugDiag:

1. 打开 DebugDiag。将打开 "选择规则类型" 对话框。选择 "故障", 然后单击 "下一步"。
2. 在 "选择目标类型" 对话框中, 选择 "特定进程", 然后单击 "下一步"。
3. 键入您的名称。EXE 转换为 "选定的进程" textbox 或从列表中选择它。如果键入, 请注意拼写。EXE 名称是否正确。请勿选中 "仅此流程实例" 框-我们希望规则对所有实例有效, 直到我们捕获到转储。完成后, 单击 "下一步"。
4. 在 "高级配置" 对话框中, 将 "最大 Userdump 限制" 设置为25, 然后单击 "例外" 按钮。
5. 单击 "添加例外"
6. 在列表中选择 "C0000005", 然后将 "操作类型" 设置为 "Full Userdump", 并将 "操作限制" 设置为25
7. 单击 "确定", 然后单击 "保存并关闭", 最后单击 "下一步"
8. 在下一个屏幕上, 请保留规则名称和路径以原样存储转储, 然后单击 "下一步"。
9. 在 "规则完成" 屏幕上, 选择 "立即激活规则", 然后单击 "完成"。现在, 当应用程序 DebugDiag 中发生故障时, 应自动创建转储文件。每次创建此规则的转储时, Userdump 计数也将在 DebugDiag IDE 内递增。注意:不需要将 "DebugDiag" 窗口保持打开状态。转储监视器实际上作为一种服务运行, 即使关闭了 DebugDiag, 也会捕获转储。

故障收集完成后, 您可以通过执行以下操作来禁用我们之前创建的规则:

1. 打开 DebugDiag。
2. 在 "规则" 选项卡中, 突出显示我们的规则, 然后右键单击并选择 "禁用规则"

C0000006 错误和疑难解答建议

2020/10/22 •

本文介绍了 C0000006 错误的含义以及如何对其进行故障排除。

原始产品版本: Visual FoxPro

原始 KB 数: 2704157

症状

您有一个用 Visual FoxPro 编写的应用程序。在报告以下错误之后, 应用程序可能会终止并关闭:

ExceptionCode: C0000006 (页面内 i/o 错误)。

原因

C0000006 错误是网络或网络连接错误。具体来说, C0000006 错误是处于页面 i/o 错误。当应用程序失去其基础网络连接时, 将发生此错误。

解决方案

由于 C0000006 错误指示网络问题, 因此最佳选择是获取网络跟踪并分析结果。这通常包括打开 Windows 平台支持组的事例, 以帮助你获取跟踪并解释结果。

当 C0000006 错误指示基础网络问题时, 您可以尝试一些建议来帮助防止此错误。

- 在 Windows 中禁用机会锁定: 在 [windows 中配置机会锁定](#)
- 将可执行文件、Visual FoxPro 运行时文件和数据文件移动到同一台计算机上。
- 请确保 VFP9RENU.dll 文件与 9 的主要内部版本 (9、8 (共 8 个) 等) 为 VFP9R.dll 和 VFP9T.dll。
- 标记。EXE \ 运行时只读。
- 禁用 (FOXUSER 创建 Foxuser 文件。DBF \。通过配置的 FPT)。带有资源的 FPW 文件 = OFF 行, 除非您特别需要该文件。Foxuser 文件是自动创建的, 并且是一个或多个文件集, 导致可能的网络故障。
- 请确保运行的是 VFP 9.0 Service Pack 2 版本。您可以从以下位置下载 Service Pack 2: [Microsoft Visual FoxPro 9.0 Service Pack 2.0](#)
- 若要生成更多想法, 请利用 VFP 论坛社区: [VFP 论坛社区](#)

在 Visual FoxPro 中捕获组合框中的输入

2020/10/22 •

本文介绍如何在 ComboBox 控件下拉列表中输入可能不包含的数据，然后将表中的字段值更改为在组合框中输入的值。

原始产品版本: Visual FoxPro

原始 KB 数: 140652

摘要

通过 Visual FoxPro, 您可以使用窗体中的组合框来选择或输入窗体中的数据。组合框允许数据派生自以下源:

- 值
- Alias
- SQL 语句
- Queries
- Arrays
- 字段
- 文件
- 结构
- 弹出窗口

本文介绍如何在 ComboBox 控件下拉列表中输入可能不包含的信息, 然后使用组合框中输入的值更改表中的字段值。

更多信息

若要將数据输入到 ComboBox 控件中, 然后将其存储到表中的字段(如果值已更改), 请按照下列步骤操作:

1. 创建一个新窗体, 并将以下代码放在其 Load 事件过程中:

```
PUBLIC ARRAY aTitle(1)
SELECT DISTINCT(customer.contact) FROM customer ;
INTO ARRAY aTitle
```

2. 将 "customer" 表从 `\Data` 目录添加到表单的数据环境中。您可以 `\Data` 在以下位置之一找到该目录:

```
Visual FoxPro 3.x: VFP\Samples\Mainsamp
Visual FoxPro 5.x: VFP\Samples
Visual FoxPro 6.0: \MSDN98\98VS\1033\Samples
Visual FoxPro 7.0,8.0,9.0: \Samples\Data
```

3. 将文本框放置在窗体上, 并将其 `ControlSource` 属性设置为:

```
Customer.Contact
```

4. 在窗体上放置一个组合框。将其 `RowSource` 属性设置为 `aTitle`, 并将其 `RowSourceType` 属性设置为 5 数组。

5. 在此 `InteractiveChange` 过程中, 键入以下代码行:

```
IF Customer.Contact != Thisform.combo1.DisplayValue
    REPLACE Customer.Contact WITH Thisform.combo1.DisplayValue
    Thisform.Text1.Refresh
    SELECT DISTINCT(customer.contact) FROM customer ;
    INTO ARRAY aTitle
ENDIF
```

6. 在此 `LostFocus` 过程中, 键入以下代码行:

```
IF Customer.Contact != Thisform.combo1.DisplayValue
    REPLACE Customer.Contact WITH Thisform.combo1.DisplayValue
    Thisform.Text1.Refresh
    SELECT DISTINCT(customer.contact) FROM customer ;
    INTO ARRAY aTitle
ENDIF
thisform.combo1.displayvalue=""
```

7. 向窗体中添加一个命令按钮, 并在其事件过程中键入以下代码 `Click` :

```
SKIP
ThisForm.Combo1.DisplayValue=Customer.Contact
ThisForm.Refresh
```

8. 向窗体中添加另一个命令按钮, 并将以下代码放在其 `Click` 事件过程中:

```
SKIP -1
ThisForm.Combo1.DisplayValue=Customer.Contact
ThisForm.Refresh
```

9. 保存并运行该窗体。更改组合框中的值, 并使用命令按钮跳过表。关闭窗口, 然后浏览 **Customer** 表。查看 " 联系人 " 字段。在组合框中输入的更改将反映在字段中。

转换函数从 SQL Server DateTime 字段中获取包含 Visual FoxPro 查询的日期

2020/10/22 •

本文介绍了如何使用 SQL Server `CONVERT()` 函数提取 Sql DateTime 字段的日期部分和 Visual FoxPro 查询。

原始产品版本: Visual FoxPro

原始 KB 数: 308133

摘要

存储在 SQL Server 表中的日期存储为日期/时间数据类型。您可以使用 SQL Server `CONVERT()` 函数, 通过 Visual FoxPro 查询提取 Sql DateTime 字段的日期部分; 例如:

```
SELECT CONVERT(CHAR(10), <field name>, 101) FROM <table name>
```

在查询中指定筛选条件时, 您仍可以只提供日期。表达式中不需要该时间。例如:

```
SELECT CONVERT(CHAR(10), <field name>, 101) FROM <table name> ;  
WHERE <field name> = '01-01-1999'
```

函数调用中的第一个和第三个参数 (CHAR (10) 和 101) 特定于从 DateTime 提取日期。CONVERT 函数的 SQL Server 联机丛书主题介绍了其他可用的选项。

更多信息

下面的 Visual FoxPro 示例代码连接到 SQL Server 并检索数据。代码使用 SQL `CONVERT()` 函数提取 DateTime 字段 "ORD_DATE" 的日期部分。转换命令返回的数据类型是 Visual FoxPro 中的字符。

若要使用此示例, 请按照下列步骤操作:

1. 将以下代码粘贴到 Visual FoxPro 中的新程序中:


```

*!* Enter your specific SQL Server information here.
#define SQL_NAME " "
#define SQL_UID " "
#define SQL_PWD " "*~~~~~

LOCAL lcSQLConnStr, ;
    lnSQLConnHandle, ;
    lnSQLExecSuccess

lcSQLConnStr = "DRIVER={SQL Server};SERVER=" + SQL_NAME + ;
";DATABASE=PUBS;UID=" + SQL_UID + ";PWD=" + SQL_PWD

lnSQLConnHandle = SQLSTRINGCONNECT(lcSQLConnStr)
IF lnSQLConnHandle < 1
    LOCAL laErrArray[1]
    AERROR(laErrArray)
    WAIT WINDOW "Unable to connect:" + CHR(13) + laErrArray[3]
    RETURN .F.
ENDIF

lnSQLExecSuccess = SQLEXP(lnSQLConnHandle, "SELECT stor_id, ord_num, ;
CONVERT(CHAR(10), ord_date, 101) AS ord_date, qty, ;
payterms, title_id from sales", "RESULTS")

*!* Note that you can still use just a date value if using a WHERE clause
*!* to filter the data:
*!* lnSQLExecSuccess = SQLEXP(lnSQLConnHandle, "SELECT stor_id, ord_num, ;
*!* CONVERT(CHAR(10), ord_date, 101) AS ord_date, qty, payterms, ;
*!* title_id from sales WHERE ord_date < '10/28/1993'", "RESULTS")

IF lnSQLExecSuccess < 1
    LOCAL laErrArray[1]
    AERROR(laErrArray)
    WAIT WINDOW "SQLEXP() Failed:" + CHR(13) + laErrArray[3]
ENDIF

SQLDISCONNECT(lnSQLConnHandle)

IF SELECT("RESULTS") > 0
    SELECT RESULTS
    BROWSE NOWAIT
ENDIF

```

2. 在指示的代码的顶部输入 SQL Server 名称、用户 ID 和密码。
3. 保存并运行代码。

参考

有关 SQL 函数的详细信息 [Convert](#)，请参阅 [Sql Server 联机丛书](#)。

在 Visual FoxPro 程序中创建主索引

2020/10/22 •

本文介绍如何在 Visual FoxPro 程序中创建主索引。

原始产品版本: Visual FoxPro

原始 KB 数: 130601

摘要

Visual FoxPro 具有新的索引类型(主)。但是,不能使用命令在程序中创建主索引 `INDEX`。本文按示例说明如何在程序中添加或修改主索引。

更多信息

主索引是永远不允许在指定的字段或表达式中使用重复值的索引。可以使用或命令在数据库容器接口或程序中建立主索引 `CREATE TABLE` `ALTER TABLE`。

带有主索引的表必须是数据库的一部分,不能是自由表。第一个示例演示如何创建包含主索引的表。第二个示例演示如何更改现有表以更改主键。第三个示例向现有表中添加主索引。

示例 1-创建包含主索引的新表

1. 创建一个新的数据库,或打开一个现有数据库。
2. 在命令窗口中发出以下命令,以在 "cSsn" 关键字字段上创建具有主索引的表:

```
CREATE TABLE employee (cSsn C(11) PRIMARY KEY, cLast C(20),;  
cFirst C(20))
```

示例 2-修改现有表以更改主索引

1. 打开示例1中使用的数据库。
2. 从命令窗口发出以下命令,将主索引从 "cSsn" 字段移动到 "cLast" 字段:

```
ALTER TABLE employee DROP PRIMARY KEY ADD PRIMARY KEY cLast TAG cLast
```

示例 3-修改现有表以添加主索引

1. 使用此命令向现有数据库添加新表:

```
CREATE TABLE noprime (cLast C(20))
```

2. 在命令窗口中,使用以下命令之一将主索引添加到现有表中:

```
ALTER TABLE noprime ADD PRIMARY KEY cLast TAG cLast
```

```
ALTER TABLE noprime ALTER COLUMN cLast C(20) PRIMARY KEY
```

在 Visual FoxPro 中创建启动屏幕

2020/10/22 •

本文介绍如何在 Visual FoxPro 中创建启动屏幕。

原始产品版本: Visual FoxPro

原始 KB 数: 138497

摘要

许多基于 Windows 的应用程序在应用程序的初始化阶段显示一个启动屏幕或启动屏幕。此启动屏幕提供了一种显示信息(如产品名称、产品徽标、产品版本信息、公司名称等)的方法。此外,它还使用户能够更快地理解应用程序启动时间。

本文提供了创建自定义应用程序启动屏幕的步骤,该屏幕将在应用程序初始化期间的指定时间间隔内显示。在可执行文件中运行时,此窗体不会禁止显示 FoxPro 系统菜单或 FoxPro 标题栏。

创建自定义启动屏幕的分步过程

您可以使用 Visual FoxPro 窗体设计器创建自定义应用程序启动屏幕。

1. 创建新表单。
2. 设置表单的以下属性:

```
AutoCenter=.T.  
BorderStyle=1  
Caption=<set the caption to be blank>  
Closable=.F.  
ControlBox=.F.  
MaxButton=.F.  
MinButton=.F.  
Movable=.F.  
WindowType=1
```

3. 向窗体中添加计时器控件。
4. 将计时器的 Interval 属性设置为启动屏幕应显示的时间长度(以毫秒为单位)。例如,如果启动屏幕显示3秒,则将计时器的 Interval 属性设置为3000。
5. 将以下代码放置在计时器的 Timer 事件中:

```
ThisForm.Release
```

6. 添加图像、标签和其他控件以根据需要自定义表单。
7. 保存并运行该窗体。

该窗体应显示在计时器的 Interval 属性中指定的时间量,然后消失。

若要将此启动屏幕合并到应用程序中,请将以下代码添加到应用程序主程序的开头:

```
DO FORM <startup screen form name>
```

现在, 当应用程序运行时, 将显示计时器的 Interval 属性中指定的时间量的启动屏幕。

在 Visual FoxPro 程序中创建存储过程

2020/10/22 •

本文介绍如何在 Visual FoxPro 程序中创建存储过程。

原始产品版本: Visual FoxPro

原始 KB 数: 130304

摘要

直观的 FoxPro 程序可以使用 CREATE DATABASE 命令创建数据库, 但 CREATE DATABASE 命令不提供创建存储过程的方法。本文介绍了如何使用 Visual FoxPro 程序将存储过程添加到新的或现有的数据库中。

更多信息

若要将存储过程添加到现有表中, 请使用 APPEND 过程命令从现有文本文件中获取过程, 并将其添加到数据库的存储过程中。

分步示例

1. 创建一个新的数据库, 或打开一个现有数据库。例如, 使用以下命令创建数据库:

```
CREATE DATABASE mydata
```

2. 创建一个文本文件, 该文件将包含所需的过程 (s) 或打开现有过程文件。文本文件应包含过程代码。如以下示例所示:

```
PROCEDURE check_stuff  
IF state <> "WA" or state <> "NC"  
WAIT WINDOW 'Invalid State'  
ENDIF  
ENDPROC
```

过程文件可能包含要在数据库的存储过程中显示的任意多个过程。

3. 若要将过程 (s) 添加到现有数据库的存储过程中, 请确保数据库已打开, 然后发出 `APPEND PROCEDURES` 命令, 如下例所示:

```
OPEN DATABASE mydata EXCLUSIVE  
APPEND PROCEDURE FROM textfile.txt && Replace textfile.txt with the  
&& actual name of your text file.
```

NOTE

APPEND 过程要求数据库以独占方式打开。此外, 还可 `APPEND PROCEDURES` 用于使用可选的 `overwrite` 子句覆盖现有存储过程, 例如:

```
APPEND PROCEDURE FROM textfile.txt OVERWRITE
```

确定正在使用的 Visual FoxPro 的版本

2020/10/22 •

本文提供了有关如何确定自定义可执行文件所使用的 Visual FoxPro 的版本的版本的信息。

原始产品版本: Visual FoxPro

原始 KB 数: 2723045

症状

您有一个 Visual FoxPro 的副本, 或者您正在运行 Visual FoxPro 中编译的自定义可执行文件。您想要确定您的 Visual FoxPro 的版本, 或者您的可执行文件正在使用。

解决方案

如果您具有 Visual FoxPro 的开发版本, 则可以使用以下任一方法来确定版本:

- 请转到 "帮助"> 关于 Microsoft Visual FoxPro。版本行报告已安装的 Visual FoxPro 的版本。
- 转到命令窗口并键入以下命令: 等待窗口版本 (# A1

如果您有一个您认为在 Visual FoxPro 中创建的自定义可执行文件, 则可以在上使用 [ListDLLs](#) 。解压. 例如, 如果您有 OurCoolApp.exe, 运行此命令将列出它已加载的 DLL: listdlls OurCoolApp。将显示与 VFP 运行时文件相关的结果输出, 如下所示:

```
0x00000000c0000000 0x486000 C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\VFP9r.dll
0x00000000d2000000 0x123000 C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\VFP9RENU.DLL
0x00000000e2000000 0x542000 C:\Program Files (x86)\Common Files\Microsoft Shared\VFP\VFP9T.DLL
```

NOTE

此输出形成一个64位的 Windows 7 计算机。因此, 您会看到 path C:\Program Files (x86) 。如果在32位计算机上运行 Visual FoxPro, 则不会有此路径。

VFP9t.dll 可能不存在, 因为只有多线程 COM 对象才需要它。

更多信息

在找到 VFP9R.dll 和 VFP9RENU.dll 的位置之后, 您可以确定其版本。打开 Windows 资源管理器并导航到 ListDLL 输出中指定的文件夹。右键单击 "", 选择 "属性", 然后查看 "详细信息" 选项卡上的 "文件版本"。

版本号左侧的第一个非零整数是 Visual FoxPro 的版本号。

使用 Visual FoxPro 将包含其他字段类型的备注字段导出到文本文件中

2020/10/22 •

本文介绍如何在逗号分隔的文本文件中将包含表中的备注字段的记录(以及其他字段类型)动态导出到一个逗号分隔的文本文件中。

原始产品版本: Visual FoxPro

原始 KB 数: 241424

摘要

本文演示如何将包含表中的备注字段的记录以及其他字段类型动态导出到逗号分隔的文本文件中。

更多信息

本文中的示例代码遵循以下顺序:

1. 提示输入源表 (.dbf 文件)。
2. 提示输入目标文本文件 (.txt 文件)。
3. 将表的内容导出到文本文件。

代码将扫描并确定表中的每个字段类型和每条记录, 然后对数据类型进行相应的处理:

```
CLOSE ALL
CLEAR ALL

lcFieldString = ''
lcMemo = ''

USE GETFILE('dbf', 'Select DBF') && Prompts for table to be used.

lnFieldCount = AFIELDS(laGetFields) && Builds array of fields from the
&& selected table.

** Prompt for Output file and use Low-Level functions
** to create it.
lcTextFile = FCREATE(GETFILE('txt', 'Select Text'))

** Starts scanning the table and converts the fields
** values according to their types **
SCAN
    WAIT WINDOW STR(RECNO()) + ' Of ' + STR(RECCOUNT()) NOWAIT

    FOR lnCount = 1 TO lnFieldCount
        lcType = laGetFields(lnCount, 2)

        IF lcType # 'G' && Don't try to turn a general field into a string
            lcString = EVALUATE(laGetFields(lnCount, 1))
        EndIf

        DO CASE
            CASE lcType = 'M' && Process the Memo Fields
                lnMemoLines = MEMLINES(EVALUATE(laGetFields(lnCount,1)))
                FOR lnLoop = 1 TO lnMemoLines
                    IF lnLoop < lnMemoLines
                        lcMemo = lcMemo + ;
```



```

        ALLTRIM(MLINE(EVALUATE(laGetFields(lnCount, 1)), ;
lnLoop)) + ' '
ELSE
    lcMemo = lcMemo + ;
        ALLTRIM(MLINE(EVALUATE(laGetFields(lnCount, 1)), ;
lnLoop))
ENDIF
ENDfor

lcString = lcMemo
lcMemo = ''
CASE lcType = 'G' && Process the General Fields
    lcString = 'Gen'
CASE lcType = 'D' && Process the Date Fields
    lcString = DTOC(lcString)
CASE lcType = 'T' && Process the DateTime Fields
    lcString = TTOC(lcString)
CASE lcType = 'N' && Process the Numeric Fields
    lcString = STR(lcString, LEN(STR(lcString)), 2)
CASE lcType = 'I' && Process the Integer Fields
    lcString = STR(lcString)
CASE lcType = 'L' && Process the Logical Fields
    IF lcString = .T.
        lcString = 'T'
    ELSE
        lcString = 'F'
    ENDIF
ENDcase

IF lnCount < lnFieldCount && Determines if the last field was
&& processed and sets the closing quote.
lcFieldString = lcFieldString + '"' + lcString + '"' + ','
ELSE
lcFieldString = lcFieldString + '"' + lcString + '"'
ENDIF
ENDfor

FPUTS(lcTextFile, lcFieldString) && Writes string to the text file.
lcFieldString = ''
ENDscan

FCLOSE(lcTextFile)

CLOSE All
CLEAR All
WAIT WINDOW 'Text File Creation Completed' NOWAIT

```

修复：尝试打印报表或编辑窗口的内容时出现错误

2020/10/22 •

本文提供了在 Citrix MetaFrame 演示服务器中使用通用打印机驱动程序功能时出现的问题的解决方案。

原始产品版本： Visual FoxPro

原始 KB 数： 909283

症状

请考虑以下方案。您使用的是 Visual FoxPro 或 Visual FoxPro 应用程序。应用程序运行在运行 Citrix MetaFrame Presentation Server 的计算机上。此外，您已将 MetaFrame 演示文稿服务器配置为使用通用打印机驱动程序功能。

当您尝试在 Visual FoxPro 中或在 Visual FoxPro 应用程序中打印报表时，您会收到以下错误消息：

错误1958:加载打印机驱动程序时出错

当您尝试在 Visual FoxPro 中打印编辑窗口的内容时，您会收到以下错误消息：

错误125:打印机未准备就绪

当您尝试在 Visual FoxPro 应用程序中打印编辑窗口的内容时，您不会收到错误消息。但是，不会打印编辑窗口中的内容。

解决方案

• Visual FoxPro 9.0

若要解决此问题，请获取 Visual FoxPro 9.0 的最新 service pack。有关详细信息，请参阅 [how to 获取 Service Pack 1 For Visual FoxPro 9.0](#)。

• Visual FoxPro 8.0

Microsoft 现在提供了受支持的修补程序，但它仅用于解决本文中所述的问题。仅适用于遇到此特定问题的系统。此修补程序可能会接受其他测试。因此，如果此问题不会对您造成严重的影响，我们建议您等待包含此修补程序的下一个 Visual FoxPro service pack。

若要立即解决此问题，请与 Microsoft 产品支持服务联系以获取修补程序。有关 Microsoft 产品支持服务电话号码的完整列表以及有关支持成本的信息，请访问以下 Microsoft 网站：

<https://support.microsoft.com/contactus/?ws=support>

NOTE

在特殊情况下，如果 Microsoft 支持专业人员确定特定更新将解决您的问题，则通常会为支持呼叫导致的费用被取消。对于不符合特定更新的其他支持问题和问题，将照常收取支持费用。

文件信息

此修补程序的英文版具有下表中列出的文件属性 (或更高版本的文件属性)。这些文件的日期和时间按协调通用时间 (UTC) 列出。当您查看文件信息时，它将转换为本地时间。若要了解 UTC 和本地时间之间的时差，请使用“控制面板”中的“日期和时间”工具中的“时区”选项卡。

Date	Time	Version	Size	File name
07-Oct-2005	01:50	8.0.0.3402	5,345,280	Vfp8.exe
07-Oct-2005	01:53	8.0.0.3402	1,466,368	Vfp8chs.dll
07-Oct-2005	01:53	8.0.0.3402	1,466,368	Vfp8cht.dll
07-Oct-2005	01:26	8.0.0.3402	1,466,368	Vfp8enu.dll
07-Oct-2005	01:53	8.0.0.3402	1,466,368	Vfp8kor.dll
07-Oct-2005	01:51	8.0.0.3402	4,300,800	Vfp8r.dll
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rchs.dll
07-Oct-2005	03:26	259,584		Vfp8rchs.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rcht.dll
07-Oct-2005	03:26	262,144		Vfp8rcht.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rcsy.dll
07-Oct-2005	03:26	269,312		Vfp8rcsy.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rdeu.dll
07-Oct-2005	03:26	270,336		Vfp8rdeu.msm
07-Oct-2005	01:30	8.0.0.3402	1,150,976	Vfp8renu.dll
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8resn.dll
07-Oct-2005	03:26	268,288		Vfp8resn.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rfra.dll
07-Oct-2005	03:26	267,776		Vfp8rfra.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rkor.dll
07-Oct-2005	03:26	261,632		Vfp8rkor.msm
07-Oct-2005	01:53	8.0.0.3402	1,150,976	Vfp8rrus.dll
07-Oct-2005	03:26	271,872		Vfp8rrus.msm
07-Oct-2005	03:25	4,206,592		Vfp8runtime.msm
07-Oct-2005	01:51	8.0.0.3402	3,776,512	Vfp8t.dll

详细信息

Citrix MetaFrame Presentation Server 3.0 和 MetaFrame Presentation Server 的更高版本包含一个名为 "通用打印机驱动程序" 的功能。您可以使用此功能处理 MetaFrame 演示文稿服务器的这些版本上的打印。

NOTE

从版本4.0 开始, 此产品称为 Citrix 演示文稿服务器。

在 Visual FoxPro 中再现问题的步骤

1. 启动 Visual FoxPro。
2. 在命令窗口中, 键入以下命令, 然后按 **enter** :

modify file printtest.txt

 将显示 "编辑" 窗口。编辑窗口包含 *Printtest.txt* 文件。但是, 此文件是空的。
3. 在编辑窗口中, 键入 *Test*。
4. 在"文件"**** 菜单上, 单击"打印"****。将显示 "打印" 对话框。
5. 在 "打印" 对话框中, 单击 "打印"。您会收到以下错误消息:

错误125:打印机未准备就绪

6. 创建一个包含以下代码 () 文件的程序, 然后运行该程序文件。

```
create table testtab1 (f1 i, f2 C(15))
for lni = 1 to 3
insert into testtab1 (f1, f2) values (lni, sys(2015))
endfor
create report printtest from testtab1.dbf
report form printtest to printer prompt
```

7. 在 " 打印 " 对话框中, 单击 " 打印"。您会收到以下错误消息:

错误1958:加载打印机驱动程序时出错

在 Visual FoxPro 应用程序中再现问题的步骤

1. 启动 Visual FoxPro。
2. 创建一个包含以下代码 () 文件的程序, 然后运行该程序文件。

```
local lcText
text to lcText noshow textmerge
local lni
* Open the editor.
modify file printtest.txt
* Create a table.
create table testtab1 (f1 i, f2 C(15))
for lni = 1 to 3
insert into testtab1 (f1, f2) values (lni, sys(2015))
endfor
* Create a report.
create report printtest from testtab1.dbf
* Print the report.
report form printtest to printer prompt
endtext
* Create a .prg file.
=StrToFile(lcText, "testxx.prg" )
* Build a project by using the .prg file.
build project testxx from testxx.prg
* Build and then run an executable file.
build exe testxx from testxx
run /n testxx.exe
return
```

Visual FoxPro 应用程序启动。当应用程序启动时, 将显示 "编辑" 窗口。

3. 在编辑窗口中, 键入 *Test*。
4. 在 " 文件 " 菜单上, 单击 " 打印 " 以打印编辑窗口的内容。
5. 在 " 打印 " 对话框中, 单击 " 打印"。您不会收到错误消息。但是, 打印作业不会打印。关闭编辑窗口。
6. 在 " 打印 " 对话框中, 单击 " 打印 " 以打印报告。您会收到以下错误消息:

错误1958:加载打印机驱动程序时出错

有关软件更新术语的详细信息, 请参阅 [用于描述 Microsoft 软件更新的标准术语的说明](#)

本文中提到的第三方产品由 Microsoft 以外的其他公司提供。对于这些产品的性能或可靠性, Microsoft 不作任何暗示保证或其他形式的保证。

设置日期或时间值以在 Visual FoxPro 中查询 SQL 表

2020/10/22 •

本文介绍如何使用远程视图或 SQL 传递查询在 Visual FoxPro 中查询 SQL Server 表中的日期或时间值。

原始产品版本: Visual FoxPro

原始 KB 数: 145817

简介

在 Visual FoxPro 中, 可以将日期类型输入数据的格式设置为 {mm/dd/yy}。或者, 如果您直接在日期字段中键入, 则可以将日期类型输入数据的格式设置为 mm/dd/yy。但是, 如果在 SQL 表上对数据进行格式设置为 {mm/dd/yy} 或 mm/dd/yy, 则会遇到 ODBC 错误。

更多信息

如果将日期值、时间值或 `timestamp` 远程查询中的值传递给 SQL 表, 则必须使用以下转义子句来设置数据格式:

- 日期值: {d 'yyyy - mm - dd'}。在此格式中, yyyy 表示年, mm 表示月, dd 表示日。
- Time 值: {t 'hh : mm : ss'}。在此格式中, hh 表示小时, mm 表示分钟, ss 表示秒。
- 时间戳值: {ts 'yyyy - mm - dd hh : mm : ss'}。

在 SQL 传递查询中, 可以使用以下语法检索日期字段的值:

```
=SQLEXEC( nConnHandle, "SELECT * FROM TITLES WHERE  
Titles.pubdate<{ts '1995-06-12 12:55:00'}", 'MyCursor')
```

您可以使用以下语法来创建查询日期和时间信息的远程视图:

```
CREATE SQL VIEW sqldate REMOTE CONNECTION sqldate AS SELECT * FROM ;  
dbo.titles WHERE Titles.pubdate<{ts '1985-06-12 12:55:00'}
```

在视图设计器中, 确保日期值、时间值或时间戳值在 "选择条件" 选项卡的 "示例" 字段中采用您希望的格式。保存视图时, 示例字段中的信息将随视图一起保存。如果打开并修改了视图, 则 "示例" 字段中的信息将丢失。如果执行此操作, 则必须重新键入信息。

NOTE

当您在远程查询中将字段名称作为参数传递而不是传递值时, 您不会遇到 ODBC 错误, 因为 Visual FoxPro 执行转换。

参考

ODBC 2.0 程序员参考和 SDK 指南, pp 为 50-52, Microsoft 新闻。

改进 Visual FoxPro 应用程序中的性能

2020/10/22 •

本文提供了有关使用 Visual FoxPro 8.0 或9.0 编写的自定义应用程序对性能问题进行改进和故障排除的建议。

原始产品版本: Visual FoxPro

原始 KB 数: 2671338

症状

您有一个自定义应用程序,它是在 Visual FoxPro 8.0 或9.0 中编写的。用户报告不符合预期的性能。本文的目标是提供改进性能的建议和一些工具,开发人员可以使用这些工具来识别导致性能瓶颈的代码区域。

原因

较差的应用程序性能 () 的根本原因很多,并且不同。它们包括(但不限于)以下各项:

- Visual FoxPro 内存使用和配置
- 软件设计 (效率低下的代码)
- 防病毒软件配置
- 本地区域网络性能
- 广域网络性能

解决方案

通常,解决任何应用程序中的性能问题,最好是使用整体方法完成。这意味着要密切检查可能影响性能的所有因素。这些问题包括使用 FoxPro 可执行文件和网络在环境中并发运行的应用程序、操作系统和软件。

本文中建议的疑难解答框架从应用程序开始。我们来看看有关设置和设计影响性能的各种因素。然后,我们会查看(服务器和本地工作站的计算机上运行的软件),最后是对网络进行考试。

更多信息

若要有效地解决 Visual FoxPro 自定义应用程序的性能较差,您需要了解操作系统环境和自定义应用程序的源代码。如果第三方供应商创建了 FoxPro 应用程序,请与他们联系以获取支持。它们具有对源代码的必要理解,这可能对解决性能问题至关重要

Visual FoxPro 内存使用和配置

您通常可以通过在应用程序)的主程序中的开头(添加一个或多个以下命令来提高 FoxPro 代码的性能:

```
SET REFRESH TO 0,0  
  
SET TABLEVALIDATE TO 2  
  
SET DELETED OFF  
  
SYS(3054) function  
  
Temp files directed locally
```

您应该单独测试任何更改,以最好地衡量对性能的影响。

软件设计 (效率低下的代码)

未优化的应用程序代码严重 hampers 应用程序性能。但是, 效率低下的代码并不总是显而易见。幸运的是, Visual FoxPro 8.0 和9.0 具有一个工具(即 "覆盖率" 探查器), 它允许您运行代码并列出行所需的时间。在开发环境中(VFP 8.0 和9.0)以及在 VFP 9.0 中的已编译可执行文件中均提供了 "覆盖范围" 探查器。

如果您具有旧版 (在 VFP 下运行的 FoxPro 2.x 过程) 代码, 则代码设计通常会成为 hampering 性能的前景色。例如, 一种可能使用定位命令而不是 SEEK 命令或查找 (# A1 函数。由于定位命令会读取表的每条记录, 因此在几乎所有情况下它通常都会降低。"覆盖范围" 探查器可帮助您确定此类瓶颈。

性能问题的另一个与代码相关的常见区域是未优化的 SELECT-SQL 命令。此外, "覆盖范围" 探查器可以确定选择-SQL 命令是否占用了大量时间。但是, 它不会告诉您如何优化查询。目标是确保选择-SQL 命令使用 **Rushmore** 最大限度的优化。

有关详细信息, 请参阅以下文章和 MSDN 链接:

- [优化应用程序](#)
- [覆盖率探查器应用程序](#)

防病毒软件配置

防病毒软件可以显著影响 VFP 应用程序的性能。尽管无人争议对防病毒软件的需求, 但如果软件扫描 VFP 数据文件, 性能可能会受到影响。虽然我们不是防病毒软件的所有可能的配置设置的专家, 但我们可以提供一些一般性建议:

禁用自动扫描将阻止防病毒软件扫描 VFP 数据文件。确保您拥有最新版本的防病毒软件。

本地区域网络性能

(LAN) 的低速局域网可能会导致应用程序性能降低。VFP 中没有允许您监视网络性能的函数或实用程序。这样做的故障排除步骤将从环境中删除 LAN 性能, 从而仅测试应用程序性能。首先, 可以从终端服务器计算机运行应用程序。应用程序在终端服务器计算机的内存空间中运行 (不是客户端操作系统) 并且只有屏幕图像需要通过局域网传输。其次, 您还可以在本地运行应用程序并消除网络。(如果只有一台客户端遇到性能下降的情况, 此选项将正常运行。)

网络问题可能很复杂, 通常包括硬件 (路由器、交换机、NIC 卡)、设置 (**Oplocks** 、 **network transmission rates**) 和软件 (操作系统版本、虚拟机)。通常, 与网络相关的问题需要涉及平台团队以获取 netmon 跟踪并运行其他操作系统诊断工具。

广域网络性能

广域网通常包含更大的地理区域和 LAN。LAN 可能连接在两个州或国家/地区内 WAN 可能连接网站的网站。随着距离增加, 更有可能导致应用程序性能瓶颈。对 Lan 的相同问题拍摄建议适用于 WAN (在终端服务器或本地测试应用程序 (如果可能))。还可能需要咨询平台团队或硬件专家。

使用 VFP OLE DB 提供程序从 VB .NET 中的 Visual FoxPro 表中查找值或最近匹配项

2020/10/22 •

本文介绍如何使用 VFP OLE DB 提供程序从 VB .NET 中的 Visual FoxPro 表中查找值或最近匹配项。

原始产品版本： Visual FoxPro

原始 KB 数： 956277

快速发布

快速发布文章可直接从 MICROSOFT 支持组织内提供信息。此处包含的信息是为响应新兴或独特的主题而创建的，或旨在补充其他知识库信息。

操作

您有一个 Visual FoxPro (VFP) 通过 VFP OLE DB 提供程序从 Visual Basic .NET 访问的表。您希望在列中查找特定值或最接近的匹配项。

解决方案

VFP SET 命令用于控制在不成功的 SEEK (或查找) 操作后，VFP 表中的记录指针会发生什么变化 (SEEK 使用 VFP 列上的索引来查找) 的值。当 NEAR 处于打开时，VFP 表中的记录指针将放置在不成功的 SEEK 之后的最接近的匹配记录。

下面的 VB .NET 代码演示如何通过 VFP OLE DB 提供程序查找记录或最接近的匹配 thereto。若要使用此代码。。。

1. 使用以下 VFP 代码创建示例 VFP 表和索引：

```
CLOSE DATA ALL
DELETE FILE C:\CUSTS.DBF RECYCLE
DELETE FILE C:\CUSTS.CDX RECYCLE
CREATE TABLE C:\CUSTS (NAMES VarChar(30))
INDEX ON UPPER(NAMES) TAG NAMES
INSERT INTO CUSTS VALUES('FRED')
INSERT INTO CUSTS VALUES('JOHN')
INSERT INTO CUSTS VALUES('MARY')
CLOSE DATA ALL
```

2. 创建新的 Visual Studio VB .NET Windows 应用程序。在窗体上放置 TextBox、CheckBox 和按钮。

3. 双击该窗体图面，打开 (form1) 的代码编辑器，然后将下面的代码粘贴到该窗体中，替换当前内容。

该代码可执行以下操作：

- 根据 VB 窗体上 CheckBox 的状态设置 (选中 = ON) 。
- 打开免费的 VFP 表 (C:\CUSTS) 并查找客户名称。
- 在 VFP OLE DB 提供程序中，创建要返回到 .NET 会话的游标。
- 使用的值填充返回游标和的 FOUND() 值 RECNO('CUSTS') RECCOUNT('CUSTS') 。
- 使用 VFP SETRESULTSET() 函数将 VFP 游标返回到 .net。
- VB .NET 使用 OleDbDataReader 对象读取返回的游标，并在中显示 SEEK 的结果 MSGBOX 。

NOTE

- 当 VFP 查找找到一个完全匹配项时, `FOUND()` (结果集内的第一列) 为 True。可以忽略其他列。
- 当 VFP 查找找不到完全匹配和 NEAR 时, 结果 `FOUND()` 集) 中的 (第一列为 False, 结果集 `RECNO()` 中的 (第二列) 将是一个有效的数字, 或者是 (`RECCOUNT()` + 1) 表示 VFP 表 (中的 EOF 的结果集内的第三个列)。
- 当 VFP 查找找不到完全匹配和 NEAR 时, `FOUND()` 将为 .f., `RECNO (# A1` 将 `RECCOUNT (# A3 + 1`。
- 结果集中的第三列是常量: `RECCOUNT('CUSTS')`。使用此来确定当 NEAR 位于最接近的记录 (即列 2 不是列 3 + 1) 时。

```
Imports System.Data.OleDb
Imports System.Text

Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        TextBox1.Focus()
        TextBox1.Text = "JOHN"
        TextBox1.CharacterCasing = CharacterCasing.Upper
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        FindOperator(TextBox1.Text, CheckBox1.Checked)
        TextBox1.Focus()
    End Sub

    Function FindOperator(ByVal Name2Find As String, ByVal Near As Boolean) As VariantType
        Dim oConn As New OleDbConnection( _"Provider=VFPOLEDB;Data Source=C:\")
        Dim oCmd As New OleDbCommand("", oConn)
        Dim oStrBldr As New StringBuilder
        Dim oReader As OleDbDataReader
        Dim lcNear As String = IIf(Near, "ON", "OFF")

        With oStrBldr
            .Append("EXECS").Append("[SET NEAR " & lcNear & "] + CHR(13) + ").Append("[USE CUSTS ORDER NAMES
            SHARED AGAIN IN 0] + CHR(13) + ").Append("[SEEK '" & Name2Find & "' IN CUSTS] + CHR(13) + ").Append("[SELECT 0]
            + CHR(13) + ").Append("[CREATE CURSOR SeekResults(lFound L, nRecno I, nReccount I)] + CHR(13) + ").Append("[
            INSERT INTO SeekResults VALUES ( FOUND('CUSTS'), RECNO('CUSTS'), RECCOUNT('CUSTS'))]+ CHR(13) + ").Append("
            [USE IN SELECT('CUSTS')] + CHR(13) + ").Append("[RETURN SETRESULTSET( 'SeekResults' )]").Append(")")
            oCmd.CommandText = .ToString
        End With

        oConn.Open()
        oReader = oCmd.ExecuteReader()
        While oReader.Read
            MsgBox("Found: " & oReader.GetBoolean(0).ToString & vbCrLf & _
                "RECNO(): " & oReader.GetInt32(1).ToString & vbCrLf & _
                "RECCOUNT(): " & oReader.GetInt32(2).ToString)
        End While

        oConn.Close()
        oConn.Dispose()
        oCmd.Dispose()
        oReader.Close()
    End Function
End Class
```

更多信息

有关本文中使用的 VFP 函数和命令的详细信息 (SEEK、NEAR、RECNO (# A2、RECCOUNT (# A4、SELECT (# A6、SETRESULTSET (# A8 等。), 请参阅 [Microsoft Visual FoxPro 9.0 SP2](#)。

VFP OLE DB 提供程序可作为免费下载, 并且是从非 VFP 应用程序访问 VFP 数据的首选方式。可在此处使用: [Visual](#)

上述

MICROSOFT 和/或其供应商对本网站上发布的文档和相关图形中包含的信息的适用性、可靠性或准确性不作任何陈述或保证, ("资料") 出于任何目的。这些材料可能包含技术上的不准确或打字错误, 并且可能随时修订, 恕不另行通知。

在适用法律允许的最大范围内, MICROSOFT 和/或其供应商不允许并排除所有表示、暗示或法定(包括但不限于)所有权、非侵权性、令人满意的条件或质量条件的表示形式、不侵权性、令人满意的条件或质量、适销性和针对特定用途的适用性(与材料相关)。

将参数传递给 SQL Server 存储过程

2020/10/22 •

本文介绍了从 Visual FoxPro 向 SQL Server 存储过程传递参数的示例。

原始产品版本: Visual FoxPro

原始 KB 数: 247370

摘要

有两种方法可以使用 SQLExec 将参数传递给存储过程。在所有版本的 Visual FoxPro 中工作的一种方法是, 将 SQL 命令生成为 string 变量。此方法的优势在于, 您可以检查字符串, 并确切查看您传递到后端的 SQL 命令。

另一种方法是传递带有问号的 Foxpro 变量, 如在参数化视图中。在 Visual FoxPro 版本5.0 及更高版本中, 这使您可以从作为输出参数返回的存储过程中获取值。

更多信息

请按以下步骤操作:

1. 在 SQL Server 中创建两个存储过程 (有关具体步骤), 请参阅联机丛书。Mysp_ObjectList 只获取 SysObjects 表, 并返回您为该表中的每条记录传递一次的值。在 mysp_GetVersion 中, 我们将详细阐述查找服务器版本的正常过程。选择 @@VERSION 通常以游标中的记录的形式返回 SQL Server 版本。在这里, 我们将此结果分配给存储过程的输出参数。

```
CREATE PROCEDURE mysp_GetVersion @tcVersion Char(200) Output AS  
SELECT @tcVersion = @@VERSION
```

```
CREATE PROCEDURE mysp_ObjectList @tcParm1 CHAR(10) AS  
SELECT @tcParm1, name FROM sysobjects
```

2. 在 ODBC 管理器中创建名为 SPParmTest 的 DSN, 它链接到您在其中创建上述过程的数据库。
3. 在 Visual FoxPro 中运行以下代码:

```
*** Error-checking is omitted for the purposes of this sample:
*** you should always check the return values from SQL Passthrough calls.
lnConn = SQLCONNECT("SPParmTest")
lcParm1 = "ReturnThis"
lcParm2 = "Then This"*** This is the first way, involving building a string
*** containing the parameters.
lcCommand = "exec mysp_ObjectList '" + lcParm1 + "'"
=SQLEXP( lnConn, lcCommand)
BROWSE
USE

*** This is the second way, passing the FoxPro variables directly to
*** the SQL command. This will work in 3.0.
lcCommand = "exec mysp_ObjectList ?lcParm2"
=SQLEXP( lnConn, lcCommand)
BROWSE
USE

*** To get a value back from a stored procedure, initialize the
*** output variable first. This won't work under 3.0.
lcVersion = SPACE(200)
lcCommand = "exec mysp_GetVersion ?@lcVersion" && Note the pass by reference.
=SQLEXP( lnConn, lcCommand)?lcVersion

=SQLDISCONNECT(lnConn) && clean up.
```

防止在 Visual FoxPro 9.0 中打印报表预览

2020/10/22 •

本文介绍了如何通过 Visual FoxPro 9.0 中使用对象辅助报告来禁用打印报告的选项。

原始产品版本: Visual FoxPro

原始 KB 数: 895279

摘要

Microsoft Visual FoxPro 9.0 引入了对象辅助报告体系结构。使用这一新设计,您可以通过代码直接通过代码进行界面,其中包含 Visual FoxPro 9.0 报告工具,如报告设计器和报告预览。因此,在 Visual FoxPro 9.0 中禁用从报告预览进行打印的功能更简单,而不是在 Visual FoxPro 的早期版本中。本文介绍如何编写代码以直接与进行交互,并配置对象辅助报告报告预览。

更多信息

默认情况下,在 Visual FoxPro 9.0 中预览对象协助的报表时,用户可以使用下列方法之一从报告预览窗口打印报告:

- 在 "报告预览" 工具栏上,单击 "打印" 按钮。
- 在报告预览窗口中右键单击,然后单击 "打印"。

您可能需要禁用此选项。在 Visual FoxPro 的早期版本中,Visual FoxPro 资源文件 (FoxUser) 必须更改。

在 Visual FoxPro 9.0 中,您仍可以将 FoxUser 文件更改为禁用 "报告预览" 窗口中的打印。但是,在引入对象辅助报告后,现在可以使用备用方法了。默认情况下,Visual FoxPro 9.0 中的对象辅助报告未启用。若要启用此设置,您必须更改 `REPORTBEHAVIOR` 设置。为此,请使用下列方法之一:

- 方法1,通过 Visual FoxPro IDE 更改报告引擎行为设置:
 1. 在 "工具" 菜单上,单击 "选项"。
 2. 在 "选项" 对话框中,单击 "报告" 选项卡,然后在 "报告引擎行为" 列表中选择 " **90 (对象协助) ** "。
 3. (可选) 如果希望此设置在 Visual FoxPro 9.0 会话之间保持,请在 "选项" 对话框中单击 "设为默认值"。
- 方法2,在代码中运行以下命令以启用对象辅助报告:

```
SET REPORTBEHAVIOR 90 .
```

代码示例

运行以下代码示例时,将预览名为 .frx 的示例报告文件。该代码示例将预览报告,并使用预览容器和预览扩展处理程序中的属性设置组合,以从报告预览窗口中完全禁用打印。

此代码示例还解决 Visual FoxPro 9.0 对象辅助报告预览中的问题,报告预览工具栏在同一报告预览会话中不能正确保留外观之间的显示设置。

此代码示例既适用于 Visual FoxPro 9.0 开发环境,也适用于使用 Visual FoxPro 9.0 创建的可执行文件,前提是您使用对象辅助报告。

若要使用此代码示例,请按照下列步骤操作:

1. 将以下代码保存到 Visual FoxPro 9.0 中的新程序文件中,然后运行代码。

```
*----- START CODE
*
*-----
```

```

* AUTHOR: Trevor Hancock
* CREATED: 02/24/05 02:47:08 PM
* ABSTRACT:
* Shows how to disable printing from
* PREVIEW when you use object-assisted
* reporting in VFP9.
*
* Also includes a workaround for problem
* where the PREVIEW toolbar does not
* recognize the TextOnToolbar or
* PrintFromPreview settings between
* toolbar displays during same preview.
*-----

*-- Defines whether to allow for
*-- printing during preview. Used to set
*-- the "AllowPrintfromPreview" property
*-- of the object-assisted preview container
*-- later in this code.
#DEFINE PrintFromPreview .T.

LOCAL loPreviewContainer AS FORM, ;
loReportListener AS REPORTLISTENER, ;
loExHandler AS ExtensionHandler OF SYS(16)

*-- Get a preview container from the
*-- .APP registered to handle object-assisted
*-- report previewing.
loPreviewContainer = NULL
DO (_REPORTPREVIEW) WITH loPreviewContainer
*-- Create a PREVIEW listener
loReportListener = NEWOBJECT('ReportListener')
loReportListener.LISTENERTYPE = 1 &&Preview

*-- Link the Listener and preview container
loReportListener.PREVIEWCONTAINER = loPreviewContainer

*-- Changing this property prevents printing from the Preview toolbar
*-- and from right-clicking on the preview container, and then clicking "print".
*-- This property is not in the VFP9 documentation at this point.
loPreviewContainer.AllowPrintfromPreview = PrintFromPreview
*-- Controls appearance of text on some of the
*-- Preview toolbar buttons. .F. is the default.
*-- Included here just to show that it is an available option.
loPreviewContainer.TextOnToolbar = .F.

*-- Create an extension handler and hook it to the
*-- preview container. This will let you manipulate
*-- properties of the container and its Preview toolbar
loExHandler = NEWOBJECT('ExtensionHandler')
loPreviewContainer.SetExtensionHandler( loExHandler )

REPORT FORM ;
HOME() + 'Samples\Solution\Reports\colors.frx' ;
OBJECT loReportListener

RELEASE loPreviewContainer, loReportListener, loExHandler

*-----
*-----
DEFINE CLASS ExtensionHandler AS CUSTOM
*-- Ref to the Preview Container's Preview Form
PreviewForm = NULL

*-- Here you implement (hook into) the PreviewForm_Assign
*-- event of the preview container's parent proxy
PROCEDURE PreviewForm_Assign( loRef )
*-- Perform default behavior: assign obj ref.
THIS.PreviewForm = loRef

```

```

*-- Grab the obj ref to the preview form and bind to its
*-- ShowToolbar() method. This lets the
*-- STB_Handler() method of this extension handler
*-- to run code whenever the Preview toolbar is shown
*-- by the PreviewForm.ShowToolbar() method.
IF !ISNULL( loRef )
BINDEVENT(THIS.PreviewForm, ;
'ShowToolbar', THIS, 'STB_Handler')
ENDIF
ENDPROC

PROCEDURE STB_Handler(lEnabled)
*-- Here you work around the setting
*-- persistence problem in the Preview toolbar.
*-- The Preview toolbar class (frxpreviewtoolbar)
*-- already has code that you can use to enforce
*-- setting's persistence; it is just not called. Here,
*-- you call it.
WITH THIS.PreviewForm.TOOLBAR
.REFRESH()
*-- When you call frxpreviewtoolbar::REFRESH(), the
*-- toolbar caption is set to its Preview form,
*-- which differs from typical behavior. You must revert that
*-- to be consistent. If you did not do this,
*-- you would see " - Page 2" appended to the toolbar
*-- caption if you skipped pages.
.CAPTION = THIS.PreviewForm.formCaption
ENDWITH
ENDPROC

*-- A preview container requires these methods
*-- to be implemented in an associated preview extension handler.
*-- They are not used in this example, but still must be here.
PROCEDURE AddBarsToMenu( cPopup, iNextBar )
PROCEDURE SHOW( iStyle )
ENDPROC
PROCEDURE HandledKeyPress( nKeyCode, nShiftAltCtrl )
RETURN .F.
ENDPROC
PROCEDURE PAINT
ENDPROC
PROCEDURE RELEASE
RETURN .T.
ENDPROC
ENDDEFINE
*
*
*----- END CODE

```

将打开 .frx 报告的预览。

NOTE

您可以通过右键单击预览或单击 "报告预览" 工具栏上的 "■" 从 "报告预览" 窗口进行打印。

2. 关闭预览, 然后在程序编辑器中修改代码。
3. 将代码顶部的 #DEFINE 语句更改为值 F。为此, 请找到以下代码。

```
#DEFINE PrintFromPreview .T.
```

将其替换为以下代码:

```
#DEFINE PrintFromPreview .F.
```

4. 保存代码, 然后再次运行它。

NOTE

现在, 无法通过右键单击预览或在报告预览工具栏上单击 "■" 从 "报告预览" 窗口进行打印。

以编程方式修改 Visual FoxPro 中的报告字段

2020/10/22 •

本文介绍如何以编程方式修改报告文件以更改字段的字体颜色。

原始产品版本: Visual FoxPro

原始 KB 数: 188403

摘要

可能需要以编程方式更改报告字段, 以根据条件来区分值或修改格式。本文演示如何基于单个销售人员的销售总额更改报表字段的字体颜色。虽然此技术在 Microsoft Visual FoxPro 9.0 专业版中起作用, 但您可以通过使用报表侦听器 and 事件驱动打印获取相同的结果。

更多信息

下面创建一个表, 填充该表并运行 "销售报表", 更改字体颜色以反映销售总计。若要更改报表字段的颜色, 请修改报表字段的记录的 PenRed、PenGreen 和 PenBlue 字段。此方法还可用于更改字体样式、字号甚至重新定位字段。此外, 还可以更改整个类的对象, 对 <property field> <property value> objtype = 8 使用 REPLACE ALL。

1. 将以下代码保存在名为 *Maketabs* 的程序文件中, 然后运行该程序以创建和填充表。

示例代码

```
*-- Code begins here.
CREATE TABLE sales (ID c(10), invamt N(8,2))
INSERT INTO sales (ID, invamt) VALUES ("JOE", 1000)
INSERT INTO sales (ID, invamt) VALUES ("MARY", 2000)
INSERT INTO sales (ID, invamt) VALUES ("HARRY", 500)
INSERT INTO sales (ID, invamt) VALUES ("JOE", 1001)
INSERT INTO sales (ID, invamt) VALUES ("MARY", 2001)
INSERT INTO sales (ID, invamt) VALUES ("HARRY", 501)
INSERT INTO sales (ID, invamt) VALUES ("JOE", 1002)
INSERT INTO sales (ID, invamt) VALUES ("MARY", 2002)
INSERT INTO sales (ID, invamt) VALUES ("HARRY", 502)
*-- Code ends here
```

2. 创建报表。在页面页眉带区中, 添加一个报告字段并将表达式设置为 salestot.id。在详细信息区段中, 添加报告字段并将表达式设置为 salestot. invamt。在页脚带区中, 添加一个报告字段并将表达式设置为 salestot. invamt。单击 "报表表达式" 对话框中的 "计算" 按钮。选择 "求和", 单击 "确定", 然后再次单击 "确定"。将报告另存为 .frx。
3. 将以下代码保存到名为 prg 的程序文件中, 并运行该程序:

```

*-- Code begins here.
SET TALK OFF
*-- Get a list of salespersons.
SELECT DISTINCT id FROM sales INTO CURSOR list

*-- Step through the list and print sales results for each
*-- salesperson.
SCAN
SELECT * FROM sales WHERE id = list.id INTO CURSOR salestot

*-- Get the total sales for this person.
SUM invamt TO lnTotal
SELECT 0

*-- Open the report file as a table.
USE sales.frx

*-- We're looking for a field (objtype) that is
*-- calculated (totaltype) and whose expression
*-- (expr) is equal to table field we're calculating.
LOCATE FOR objtype = 8 AND totaltype = 2 AND ;
ATC("salestot.invamt", expr) > 0

*-- Save the values for the pen and fill fields.
lnPenRed = penred
lnPenGreen = pengreen
lnPenBlue = penblue

*-- Modify the font color for the sales total.
DO CASE
CASE lnTotal > 3000 AND lnTotal < 6000 && Print green
REPLACE penred WITH 0, pengreen WITH 255, penblue WITH 0
CASE lnTotal > 6000 && Print blue
REPLACE penred WITH 0, pengreen WITH 0, penblue WITH 255
OTHERWISE && Print dark red on white
REPLACE penred WITH 128, pengreen WITH 0, penblue WITH 0
ENDCASE
USE
SELECT salestot

*-- Change PREVIEW to TO PRINTER to print the report.
REPORT FORM sales PREVIEW
ENDSCAN()

*-- Restore the original values.
USE sales.frx
LOCATE FOR objtype = 8 AND totaltype = 2 AND ;
ATC("sales.invamt", expr) > 0
REPLACE penred WITH lnPenRed, pengreen WITH lnPenGreen,;
penblue WITH lnPenBlue
USE
CLOSE DATA
*-- Code ends here.

```

4. 在预览每个报表时，滚动到底部以查看摘要，并观察每个销售人员的颜色变化情况。

如果日期格式不是美国日期格式，则 Visual FoxPro ODBC 驱动程序查询将失败

2020/10/22 •

本文帮助您解决使用 Visual FoxPro ODBC 驱动程序对 Visual FoxPro 表运行 ODBC 查询时出现的问题。

原始产品版本： Visual FoxPro

原始 KB 数： 229854

症状

使用 Visual FoxPro ODBC 驱动程序对 Visual FoxPro 表执行 ODBC 查询时，如果 WHERE 子句包含不采用美国日期格式的日期，则不返回任何记录。

原因

Visual FoxPro ODBC 驱动程序仅接受全美日期格式的日期。

解决方案

将 SELECT-SQL 语句中的 WHERE 子句中传递的任何日期转换为美国日期格式。

状态

此行为是设计使然。

更多信息

默认的 Visual FoxPro 日期设置为美国。但是，日期格式可能设置为以下格式：

格式	DATE FORMAT
美式	mm/dd/yy
ANSI	yy.mm.dd
英国/法语	dd/mm/yy
德语	dd mm
意大利语	dd-mm-yy
日本	yy/mm/dd
台湾	yy/mm/dd
特区	mm-年月日
MDY	mm/dd/yy

日期格式	DATE FORMAT
DMY	dd/mm/yy
YMD	yy/mm/dd
SHORT	由 Windows 控制面板短日期设置确定的短日期格式。
大量	由 Windows 控制面板长日期设置确定的长日期格式。

再现行为的步骤

1. 使用以下代码创建名为 *Odbctest* 的程序文件：

```
CLEAR
DO CASE
    CASE "6.0"$VERSION()
        lcConnStr="DRIVER={Microsoft Visual FoxPro Driver};" + ;
        "Exclusive=No;SourceType=DBF;SourceDB="+HOME(2)+"DATA"
    CASE "5.0"$VERSION()
        lcConnStr="DRIVER={Microsoft Visual FoxPro Driver};" + ;
        "Exclusive=No;SourceType=DBF;SourceDB="+HOME()+"SAMPLES\DATA"
    CASE "3.0"$VERSION()
        lcConnStr="DRIVER={Microsoft Visual FoxPro Driver};" + ;
        "Exclusive=No;SourceType=DBF;SourceDB="+HOME()+"SAMPLES\DATA"
    OTHERWISE && Version is VFP 7.0,8.0, or 9.0
        lcConnStr="DRIVER={Microsoft Visual FoxPro Driver};" + ;
        "Exclusive=No;SourceType=DBF;SourceDB="+HOME(2)+"DATA"
ENDCASE

*!* Create An ADO Connection

oConnection=CREATEOBJECT("ADODB.Connection")
oConnection.ConnectionString = lcConnStr
oConnection.CursorLocation = 3
oConnection.OPEN

* lcSQL="SELECT * FROM ORDERS WHERE ORDER_DATE < {07/22/93}"
lcSQL="SELECT * FROM ORDERS WHERE ORDER_DATE < {93/07/22}"*!* Create An ADO recordset
rs=CREATEOBJECT("ADODB.Recordset")
rs.activeconnection = oConnection
rs.CursorLocation = 3
rs.cursortype = 1
rs.LockType = 3
rs.OPEN(lcSQL)
IF !rs.EOF
    rs.movefirst
    DO WHILE !rs.EOF
        ? rs.FIELDS(0).VALUE
        rs.movenext
    ENDDO
ENDIF
rs.CLOSE
oConnection.CLOSE
```

2. 请注意，不会返回或显示任何记录。

3. 对以下代码行添加注释：

```
lcSQL="SELECT * FROM ORDERS WHERE ORDER_DATE < {93/07/22}"
```

4. 取消注释以下代码行：

```
lcSQL="SELECT * FROM ORDERS WHERE ORDER_DATE < {07/22/93}"
```

5. 重新运行程序并观察数据返回并显示在屏幕上。

在 Visual FoxPro 中使用 MScomm32 从串行端口接收

2020/10/22 •

本文介绍了在 Visual FoxPro 中使用 Mscomm32 控件从串行端口接收数据的两种方法。

原始产品版本: Visual FoxPro

原始 KB 数: 140525

摘要

本文为您提供了两种技术, 可用于通过使用 Mscomm32 控件从串行端口接收数据。第一种方法使用事件驱动的方法, 不要求您轮询串行端口以检查是否存在收到的字符。此技术可以最大限度地提高灵活性, 并且不需要大量编码来阻止缓冲区运行。第二种方法要求您定期轮询输入缓冲区, 以检查是否存在收到的字符。本文介绍了这两种方法, 并提供了每种方法的示例。

更多信息

无论您使用哪种技术, 都可以设置以下属性:

- CommPort** 属性: 将此数值属性设置为所需的通信端口。有效值为1、2、3或4, 具体取决于可用的串行端口和各个计算机的配置。这些值分别对应于 Com1、Com2、Com3 和 Com4。
- Settings** 属性: 将此字符属性设置为连接到串行端口的设备所需的波特率、奇偶校验、数据位和停止位。此属性是一个字符(以逗号分隔的列表)。例如, 若要将串行端口设置为14400波特、偶校验、7数据位和1停止位, 请将字符串设置为: `14400,E,7,1`。
- PortOpen** 属性: 将此逻辑属性设置为 true 可打开与串行端口的通信。您还可以检查此属性以确定端口是否已正确打开。

技术一: 事件驱动的接收

OnComm 当输入缓冲区中存在等待的字符时, 事件驱动技术将生成事件。此外, CommEvent 属性将包含一个数字2。
OnComm 若要触发该事件, 必须将该属性设置 **Rthreshold** 为非零值(其默认)。该属性最常见的设置 **Rthreshold** 为1, 这意味着, **OnComm** 如果在输入缓冲区中至少有一个字符在等待, 则会触发事件。

例如, 您可以将以下代码放在事件中, **OnComm** 以将接收到的数据追加到名为的表单的属性中 **mybuffer** :

```
Procedure MyCom.OnComm
  IF This.CommEvent = 2
    ThisForm.mybuffer = ThisForm.mybuffer + This.Input
  ENDIF
ENDPROC
```

技术二: 轮询输入缓冲区

轮询输入缓冲区需要程序定期停止正在执行的操作, 并检查是否有在输入缓冲区中等待的字符。使用此技术时, 将 **Rthreshold** 属性保留为0(其默认值), 并检查 **InBufferCount** 属性以查看它是否大于零, 这表示缓冲区中存在等待的字符。

NOTE

使用诸如检查输入属性的长度导致丢失字符这样的技术将导致丢失的字符,这是因为只要访问了 Input 属性,输入缓冲区就会被清空。请改为使用 InBufferCount 属性。

假设 Mscomm 控件在窗体上且具有名称 `MyCom` 且有一个名为的窗体属性 `mybuffer`, 以下代码说明了如何轮询等待字符:

```
Procedure myform.myproc
  IF Thisform.MyCom.InBufferCount > 0
    Thisform.mybuffer = Thisform.mybuffer + Thisform.MyCom.Input
  ENDIF
ENDPROC
```

可以在 timer 方法中调用过程代码, 以便于以半常规间隔检查字符。但是, 如果预计从串行端口收到大量数据, 技术 1 将降低在输入缓冲区运行的可能性。

在运行时，在 Visual FoxPro 中更新游标时，刷新的网格不显示任何数据

2020/10/22 •

本文提供了这样一种问题的解决方案：在使用 ThisForm 或 ThisForm 命令时，将 RecordSource 属性设置为游标的网格不会刷新数据，即使您使用的是 "" 或 "GridName" 命令也是如此。

原始产品版本： Visual FoxPro

原始 KB 数： 131836

症状

在 Visual FoxPro 中，如果 RecordSource 将网格的属性设置为游标，然后在运行时使用命令更新光标

SQL Select Into Cursor CursorName，则网格不会显示任何数据，即使您使用 ThisForm.Refresh 命令或命令刷新它也不会显示任何数据 ThisForm.GridName.Refresh。

原因

当您执行 SQL Select 语句到在网格的属性中指定的同一游标时 RecordSource，必须先销毁原始游标，然后才能创建新游标。发生这种情况时，网格列和的列 RecordSource 也将被清除，然后重新创建。将网格的记录源设置为新游标，网格将自动创建列。新的网格从零开始工作。因此，它不会从游标中加载数据。

解决方案

若要在这种情况下刷新网格，请将 RecordSource 网格的属性设置为其自身，如下例所示：

```
thisform.grid1.recordsource=thisform.grid1.recordsource
```

状态

此行为是设计使然。

更多信息

有关详细信息，请参阅 [如果选择-SQL 命令在 Visual FoxPro 中重置光标，则基于游标的网格是空的。](#)

再现行为的步骤

1. 创建一个窗体，然后添加该客户。DBF 表 (\VFP\SAMPLES\DATA) 到数据环境的目录中。
2. 将以下代码放在窗体的 Load 事件中：

```
CREATE CURSOR Compdisp (company c(40),city c(15),country c(15))
```

3. 将以下代码放在表单的销毁事件中：


```
SELECT Compdisp  
USE
```

4. 在窗体上创建一个网格，并为网格指定以下属性设置：

```
ColumnCount=3  
RecordSourceType=Alias  
RecordSource=Compdisp
```

5. 向窗体中添加一个文本框。

6. 在其事件中添加带有以下代码的命令按钮 `Click`：

```
SELECT company,city,country;  
FROM customer;  
WHERE customer.country=thisform.text1.value;  
INTO CURSOR Compdisp  
THISFORM.GRID1.REFRESH
```

7. 运行窗体。在文本框中键入 `华北`，然后单击命令按钮。网格应显示所有记录 `Customer.Country="France"`，但不显示。网格为空。`THISFORM.GRID1.REFRESH` 似乎不更新网格。即使在窗体的 `Load` 事件中不创建游标，也会发生这种情况。

- 若要使网格正确显示数据，请将以下命令作为命令按钮事件的最后一行代码 `Click`：

```
ThisForm.Grid1.RecordSource =ThisForm.Grid1.RecordSource
```

手动注册 ActiveX 控件 (.ocx)

2020/10/22 •

本文介绍如何使用 Regsvr32 命令手动注册 ActiveX 控件。

原始产品版本: Visual FoxPro

原始 KB 数: 146219

摘要

当您分发使用 ActiveX 控件 (.ocx 文件) 的 Microsoft Visual FoxPro 应用程序时, 必须正确注册 .ocx 文件以使其正常工作。Visual foxpro 安装向导或 visual foxpro 7.0 中的 InstallShield Express, 或 Visual FoxPro 的更高版本将正确注册 .ocx 文件, 前提是您在步骤6中选中了该 .ocx 文件的 "OLE " 复选框。如果使用 .ocx 文件的 Visual FoxPro 应用程序是通过其他一些方法分发的, 则必须手动注册 .ocx 文件。本文介绍如何手动注册 .ocx 文件。

更多信息

您可以使用 Microsoft Register Server (# A0) 在32位操作系统上手动注册32位 .ocx 文件。在 Visual FoxPro 3.0 和 3.0 b 中, Regsvr32.exe 位于 \Vfp\Samples\Ole 目录中, 在 Visual FoxPro 5.0 中, Regsvr32.exe 位于 \Vfp 目录中。在 Visual FoxPro 6.0 中, 在 Visual FoxPro 目录的分布目录中找到 Regsvr32.exe。它可以通过 Visual FoxPro 应用程序分发。使用 Regsvr32.exe 的语法如下所示:

```
Regsvr32 [/u] [/s] <OCX File Name>
```

NOTE

/u 表示注销 .ocx 文件。/s 表示静默模式 () 中不显示任何邮件。

下面的示例在不显示任何邮件的情况下注册 Microsoft MAPI ActiveX 控件:

```
Regsvr32 /s MSMAPI32.OCX
```

若要在 Visual FoxPro 应用程序中实现此示例, 请使用 "运行" 命令, 如下所示:

```
RUN /N Regsvr32 /s MSMAPI32.OCX
```

NOTE

如果在注册控件时发生错误, 请执行以下操作:

1. 确认注册表中的控件以前尚未注册过。
2. 验证 Windows\System 目录中是否有以下文件:

```
mfc30.dll olepro32.dll msvcrt20.dll  
mfc40.dll msvcrt40.dll
```

如果缺少其中一个文件，您可能会收到以下错误消息：

错误：OLE 错误代码0x80040112：找不到此类的相应许可证。

使用 Mscomm32 发送到串行端口

2020/10/22 •

本文介绍如何使用 Mscomm32 控件将信息发送到串行端口。

原始产品版本: Visual FoxPro

原始 KB 数: 139526

摘要

本文介绍了使用 Mscomm32 将数据发送到串行端口所需的设置。Mscomm32 附带 Microsoft Visual FoxPro Professional Edition。它可以在运行 Microsoft Windows 95 和更高版本 Windows 的计算机上使用。

更多信息

使用 mscomm 控件将数据发送到串行端口的最常用的属性如下所示:

CommPort

CommPort 属性指定通信端口号。一个与 Comm 端口相对应的数值属性。默认情况下, 此属性设置为1对应于 com1。有效值为1、2、3或4, 具体取决于安装在计算机上的串行端口及其配置。

设置

Settings 属性为串行端口配置波特率、奇偶校验、数据位和停止位。Settings 属性是包含单个以逗号分隔的值的字符串。默认情况下, Settings 属性如下所示: 9600、N、8、1 此属性对应于9600波特、无奇偶校验、8个数据位和1个停止位。

以下波特率值有效: 110、300、600、1200、2400、4800、9600 (默认)、14400、19200、28800、38400、56000、57600、115200、128000、256000。

PortOpen

PortOpen 属性指定一个逻辑值, 控制串行端口是否打开并处于活动状态。将以前的属性设置为开始使用串行端口后, 可以将该属性设置为 true。

Output

将为 Output 属性分配要发送到串行端口的字符串。若要在设置以前的属性后将字符串 "Hello World" 输出到串行端口, 请使用以下命令:

```
myform.mycomm.output = "Hello World"
```

还可能需要的其他属性取决于应用程序, 如下所示:

CommEvent

CommEvent 属性包含一个值, 该值表示最新的通信事件或错误。

Sthreshold

该 **Sthreshold** 属性指定发送的输出缓冲区中的最小字符数。

OutBufferCount

OutBufferCount 控件返回在传输缓冲区中等待的字符数。如果属性为零, 则此值应始终为零 **Sthreshold**。将此

OutBufferCount 属性设置为零将清除传输缓冲区。

OutBufferSize

OutBufferSize 属性指定传输缓冲区的大小。默认情况下, 此缓冲区为512字节。传输缓冲区越大, 其他应用程序可用的内存越少。较慢的波特率和向串行端口写入的大型文本字符串可能意味着您需要更大的值。

下面的示例展示了如何设置 comm 控制并使用标准 Hayes 调制解调器命令拨打电话号码555-1234。

```
PUBLIC ComForm
ComForm = CREATEOBJECT('Form')
ComForm.AddObject("Testcom", "Olecontrol", "MSCOMMLib.MSComm")
ComForm.Testcom.CommPort = 2 && Use Comm2, The second Serial Port.
ComForm.Testcom.Settings = "14400,N,8,1" && 14.4 Kbaud, No Parity,
&& 8 data Bits, 1 Stop Bit
ComForm.Testcom.PortOpen = .T.
ComForm.Testcom.Output = "ATDT555-1234" + chr(13) && Dialing the number
* The chr(13) is needed to complete the modem command sequence
ComForm.Testcom.PortOpen = .F.
***** End Code *****
```

NOTE

如果在应用程序中使用此方法, 然后使用 Visual FoxPro 安装向导分发应用程序, 则在运行该应用程序时, 可能会看到以下错误消息:

程序错误 OLE 错误代码0x80040112: 找不到此类的相应许可证。

如何获取 Visual FoxPro 9.0 的 Service Pack 1

2020/10/22 •

本文介绍有关 Visual FoxPro 9.0 Service Pack 1 的详细信息。

原始产品版本: Visual FoxPro

原始 KB 数: 906478

介绍

Visual FoxPro 9.0 更新分布在 service pack 中。Service pack 有助于保持 Visual FoxPro 9.0 电流。Service pack 还扩展并更新了相关产品的功能。此外, service pack 扩展了计算机的功能。服务包是累积的。每个新 service pack 都包含以前的 service pack 和任何新修补程序附带的所有修补程序。在安装最新版本之前, 无需安装以前的 service pack。

Visual FoxPro 9.0 Service Pack 1

发布日期: 2005年12月8日

如何获取

若要下载 Visual FoxPro 9.0 Service Pack 1 (SP1), 请参阅 [下载中心](#)

修复列表

有关 Visual FoxPro 9.0 Service Pack 1 中包含的修补程序列表的详细信息, 请参阅 [Microsoft Visual foxpro 9.0 Service pack 1 中修复的 Bug 列表](#)。

参考

有关软件更新术语的详细信息, 请参阅 [用于描述 Microsoft 软件更新的标准术语的说明](#)。

使用 FoxPro 文件进行 SQL Server 分布式查询

2020/10/22 •

本文介绍如何使用 VFP ODBC 驱动程序或 VFP OLE DB 提供程序执行 SQL Server 分布式查询，以检索 dbc 和 .dbf 文件中的数据。

原始产品版本： Visual FoxPro

原始 KB 数： 207595

摘要

本文演示如何使用 `.dbc` `.dbf` 使用 VFP ODBC 驱动程序或 VFP OLE DB 提供程序执行 SQL Server 分布式查询，以检索 FoxPro 和文件中的数据。

更多信息

Microsoft SQL Server 2000 提供了对 OLE DB 提供程序执行查询的功能。此查询通过使用 `OpenQuery` 或 `OpenRowset` `transact-sql` 函数或使用包含四部分名称(包括链接服务器名称)的查询来完成。

例如：

```
sp_addlinkedserver 'mylinkedserver', 'product_name', 'myoledbprovider', 'data_source','location',
'provider_string', 'catalog'

SELECT * FROM OPENQUERY(mylinkedserver, 'select * from table1')
```

您应使用 Microsoft OLE DB provider for ODBC (MSDASQL) 和 Visual FoxPro ODBC 驱动程序来设置链接服务器，以对 FoxPro 和文件执行分布式查询 `.dbc` `.dbf`。不支持将 Jet OLEDB 提供程序与 FoxPro 一起使用。VFP ODBC 驱动程序不是线程安全的。由于 SQL Server 是多线程的，在某些情况下，VFP ODBC 驱动程序可能会出现問題。如果可能，我们建议使用 VFP OLE DB 提供程序连接到 SQL Server 数据。

下面的 T-sql 代码示例演示如何设置和使用带有 `OpenQuery` 和 `OpenRowset` 函数的 FoxPro 的分布式查询。它还演示了如何从 SQL Server 2000 更新远程 FoxPro 表。在 sql Server 2000 计算机上安装 Visual FoxPro ODBC 驱动程序后，可以在 SQL 查询分析器中测试此代码。您需要将数据源名称和路径更改为相应的 FoxPro 文件：

```
/* OPENROWSET and OPENQUERY examples with VFP via ODBC OLE DB provider */

/* These OPENROWSET examples depend on the sample files VFP98\data\Testdata.dbc
Modify your code accordingly for differences in location or DBC name */

=====
-- Using DBC file , read and update
=====
-- OPENROWSET DSN-less example

select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data\Testdata.dbc;
SourceType=DBC',
'select * from customer where country != "USA" order by country')
go

select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data\Testdata.dbc;
```

```

SourceType=DBC',
'select * from customer where region="WA"')
go

Update openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data\Testdata.dbc;
SourceType=DBC',
'select * from customer where region="WA"')
set region = "Seattle"
go

-- check to verify which rows were updated
select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data\Testdata.dbc;
SourceType=DBC',
'select * from customer where region="Seattle"')
go

-- OPENROWSET DSN example
/* Note the DSN Example might fail if SQL Server is configured to use a local account.*/
select * from openrowset('MSDASQL',
'DSN=Visual FoxPro Database;
SourceDB=e:\VFP98\data\Testdata.dbc;
SourceType=DBC',
'select * from customer where country != "USA" order by country'
go

/* sp_addlinkedserver examples */
-- sp_addlinkedserver example with DSN

/* You will need to make a DSN and point it to the Testdata database.
Modify your code accordingly for differences in location or DBC name */

/* Note this Example may fail if SQL Server is configured to use a local account.*/
sp_addlinkedserver 'VFP Testdata Database With DSN',
'',
'MSDASQL',
'VFP System DSN'
go

sp_addlinkedsrvlogin 'VFP Testdata Database With DSN', FALSE, NULL, NULL, NULL
go

SELECT *
FROM OPENQUERY([VFP Testdata Database With DSN], 'select * from customer where region = "Seattle"')
go

-- Update using OpenQuery
Update OPENQUERY([VFP Testdata Database With DSN], 'select * from customer where region="WA"')
set region = &quot;Seattle&quot;
go

/* SP_addlinkedserver example with DSN-less connection */

/* This example also depends on the sample files Testdata.dbc
Modify your code accordingly for differences in location or DBC name */

sp_addlinkedserver 'VFP Testdata Database With No DSN',
'',
'MSDASQL',
NULL,
NULL,
'Driver={Microsoft Visual FoxPro
Driver};UID=;PWD=;SourceDB=e:\VFP98\data\Testdata.dbc;SourceType=DBC;Exclusive=No;BackgroundFetch=Yes;Collate=M
achine;'
go

```



```

sp_addlinkedserver 'VFP Testdata Database With No DSN', FALSE, NULL, NULL, NULL
go

SELECT *
FROM OPENQUERY([VFP Testdata Database With No DSN], 'select * from customer where country != "USA" order by
country')
go

-----
-- Using VFP 6.0 driver, read and update data from VFP sample dbf files
-----

-- OPENROWSET DSN-less example

select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data;
SourceType=DBF',
'select * from customer where country != "USA" order by country')
go

-- perform UPDATE

Update openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data;
SourceType=DBF',
'select * from customer where region="Seattle"')
set region = "WA"
go

-- verify update

select * from openrowset('MSDASQL',
'Driver=Microsoft Visual FoxPro Driver;
SourceDB=e:\VFP98\data;
SourceType=DBF',
'select * from customer where region = "WA"')
go<br/>

-- OPENROWSET DSN example
-- DSN points to the folder where .dbf files are.
/* Note this Example may fail if SQL Server is configured to use a local account.*/
select * from openrowset('MSDASQL',
'DSN=Visual FoxPro Tables;
SourceDB=e:\VFP98\data;
SourceType=DBF',
'select * from customer where country != "USA" order by country')
go"?
-- SQL Server's QUOTED_IDENTIFIER has to be set to OFF.

SET QUOTED_IDENTIFIER OFF

-- OPENROWSET DSN-less example

select * from openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP98\samples\data\Testdata.dbc;

SourceType=DBC',

'select * from customer where country = "USA" order by city')

go

select * from openrowset('MSDASQL',

```

```

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data\Testdata.dbc;

SourceType=DBC',

'select * from customer where region="WA"')

go

Update openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data\Testdata.dbc;

SourceType=DBC',

'select * from customer where city = "Seattle"')

set region = "WW"

go

-- check to verify which rows were updated

select * from openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data\Testdata.dbc;

SourceType=DBC',

'select * from customer where region="WW"')

go

-- OPENROWSET DSN example

/* Note the DSN Example might fail if SQL Server is configured to use a local account.*/

select * from openrowset('MSDASQL',

'DSN=Visual FoxPro Database;

SourceDB=e:\VFP90\samples\data\Testdata.dbc;

SourceType=DBC',

'select * from customer where country = "USA" order by city')

go

/* sp_addlinkedserver examples */

-- sp_addlinkedserver example with DSN

/* You will need to make a DSN and point it to the Testdata database.

Modify your code accordingly for differences in location or DBC name */

/* Note this Example may fail if SQL Server is configured to use a local account.*/

sp_addlinkedserver 'VFP Testdata Database With DSN',

'',

'MSDASOL'.

```

```
'VFP System DSN'
```

```
go
```

```
sp_addlinkedserver 'VFP Testdata Database With DSN', FALSE, NULL, NULL, NULL
```

```
go
```

```
SELECT *
```

```
FROM OPENQUERY([VFP Testdata Database With DSN], 'select * from customer where city = "Seattle" ')
```

```
go
```

```
-- We will set the region back to "WA" if it currently is "WW".
```

```
-- Update using OpenQuery
```

```
Update OPENQUERY([VFP Testdata Database With DSN], 'select * from customer where city = "Seattle" ')
```

```
set region = "WA"
```

```
go
```

```
-- Make sure that the region got updated.
```

```
SELECT *
```

```
FROM OPENQUERY([VFP Testdata Database With DSN], 'select * from customer where city = "Seattle" ')
```

```
go
```

```
/* SP_addlinkedserver example with DSN-less connection */
```

```
/* This example also depends on the sample files Testdata.dbc
```

```
Modify your code accordingly for differences in location or DBC name */
```

```
sp_addlinkedserver 'VFP Testdata Database With No DSN',
```

```
'',
```

```
'MSDASQL',
```

```
NULL,
```

```
NULL,
```

```
'Driver={Microsoft Visual FoxPro
```

```
Driver};UID=;PWD=;SourceDB=e:\VFP90\samples\data\Testdata.dbc;SourceType=DBC;Exclusive=No;BackgroundFetch=Yes;Collate=Machine;'
```

```
go
```

```
sp_addlinkedserver 'VFP Testdata Database With No DSN', FALSE, NULL, NULL, NULL
```

```
go
```

```
SELECT *
```

```
FROM OPENQUERY([VFP Testdata Database With No DSN], 'select * from customer where country = "USA" order by city')
```

```
go
```

```
=====
```

```
-- Using VFP 6.0 driver, read and update data from VFP sample dbf files
```

```

-- Using VFP 9.0 driver, read and update data from VFP sample DBF files

=====

-- OPENROWSET DSN-less example

select * from openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data;

SourceType=DBF',

'select * from customer where country != "USA" order by country')

go

-- perform UPDATE

Update openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data;

SourceType=DBF',

'select * from customer where city = "Seattle"')

set region = "WW"

go

-- verify update

select * from openrowset('MSDASQL',

'Driver=Microsoft Visual FoxPro Driver;

SourceDB=e:\VFP90\samples\data;

SourceType=DBF',

'select * from customer where region = "WW"')

go

-- OPENROWSET DSN example

-- DSN points to the folder where .dbf files are.

/* Note this Example may fail if SQL Server is configured to use a local account.*/

select * from openrowset('MSDASQL',

'DSN=Visual FoxPro Tables;

SourceDB=e:\VFP90\samples\data;

SourceType=DBF',

'select * from customer where country != "USA" order by country')

go

```

您还可以使用 Visual FoxPro OLE DB 提供程序来创建分布式查询。它是要使用的首选技术。虽然此代码演示了如何更新和删除数据，但不支持使用 OLE DB Provider 在分布式查询中添加、更新 (编辑) 和删除数据。

下面的 T-sql 代码示例演示如何设置和使用带有 OpenQuery 和函数的 FoxPro 的分布式查询 `OpenRowset`。在 sql Server 2000 计算机上安装 Visual FoxPro OLE DB 提供程序后, 可以在 SQL 查询分析器中测试此代码。您需要将数据源名称和路径更改为相应的 FoxPro 文件:

```
/* These OPENROWSET examples depend on the sample files VFP98\data\Testdata.dbc

'Modify your code accordingly for differences in location or DBC name */

--*=====

--* Using the DBC file, reading and updating data.

--*=====

--* A couple of OPENROWSET queries.

select * from openrowset('VFPOLEDB',

'e:\vfp7junk\Testdata.dbc';'Exclusive=No';'Data Source=DBC',

'select * from customer where country != "USA" order by country')

go

Select * from openrowset('VFPOLEDB',

'e:\vfp7junk\Testdata.dbc';'Exclusive=No';'Data Source=DBC',

'select * from customer where region="WA"')

go

--* Need to use an error trapping routine with the UPDATE and DELETE functions:

select * from

openrowset('VFPOLEDB',

'E:\VFP7Junk\Testdata.DBC';'Exclusive=No';'Data Source=DBC',

'Update Customer Set city = "SEATTLE" where region = "WA" ')

go

declare @upderror int

select @upderror = @@error

print ''

if @upderror != 7357 and @upderror != 0

print 'Update failed with error '+convert(varchar(5),@upderror)

else

print 'Ignore the error above, the Update succeeded'

go

-- check to verify which rows were updated

select * from openrowset('VFPOLEDB',

'E:\VFP7junk\Testdata.DBC';'Exclusive=No';'Data Source=DBC',

'select * from customer where region = "WA"')
```

```

go

--* Change the City field back to "Seattle".

select * from

openrowset('VFPOLEDB',

'E:\VFP7Junk\Testdata.DBC';'Exclusive=No';'Data Source=DBC',

'Update Customer Set city = "Seattle" where region = "WA" ')

go

declare @upderror int

select @upderror = @@error

print ''

if @upderror != 7357 and @upderror != 0

print 'Update failed with error '+convert(varchar(5),@upderror)

else

print 'Ignore the error above, the Update succeeded'

go

--* The DELETE fucntion also causes an error, but the DELETE works.

select * from

openrowset('VFPOLEDB',

'E:\VFP7Junk\Testdata.DBC';'Exclusive=No';'Data Source=DBC',

'Delete from Customer where country = "Spain" ')

go

declare @delerror int

select @delerror = @@error

print ''

if @delerror != 7357 and @delerror != 0

print 'Delete failed with error '+convert(varchar(5),@delerror)

else

print 'Ignore the error above, the Delete succeeded'

go

--* Check to see that the records are deleted.

Select * from openrowset('VFPOLEDB',

'e:\vfp7junk\Testdata.dbc';'Exclusive=No';'Data Source=DBC',

'select * from customer where country = "Spain"')

go

```

```

--* Here are some examples using the VFP OLE DB Provider to create Linked Servers.

--* Using sp_addlinkedserver to create the Linked Server.

sp_addlinkedserver @server='VFP_Linked_Server',

@srvproduct='Microsoft Visual FoxPro OLE DB Provider',

@provider='VFPOLEDB',

@datasrc = 'E:\vfp7junk'

go

SELECT *

FROM OPENQUERY([VFP_Linked_Server], 'select * from customer where city = "Seattle"')

go

-- The Update command will update the table with the OPENQUERY function when using the

-- linked server, but the same error 7357 error will occur.

select * from

OPENQUERY([VFP_Linked_Server],

'Update Customer Set city = "SEATTLE" where region = "WA" ')

go

declare @upderror int

select @upderror = @@error

print ''

if @upderror != 7357 and @upderror != 0

print 'Update failed with error '+convert(varchar(5),@upderror)

else

print 'Ignore the error above, the Update succeeded'

go

-- Check and see if the City field is all uppercase with "SEATTLE".

SELECT *

FROM OPENQUERY([VFP_Linked_Server], 'select * from customer where region = "WA"')

go

--* Let's check for how many records have the word "London" in the City field.

SELECT *

FROM OPENQUERY([VFP_Linked_Server], 'select * from customer where city = "London"')

go

-- We can also use the Delete command to remove records with the OPENQUERY function when using the

-- linked server, but the same error 7357 error will occur.

select * from

```

```

OPENQUERY([VFP_Linked_Server],

'Delete from Customer where city = "London"')

go

declare @delerror int

select @delerror = @@error

print ''

if @delerror != 7357 and @delerror != 0

print 'Delete failed with error '+convert(varchar(5),@delerror)

else

print 'Ignore the error above, the Delete succeeded'

go

/* SP_addlinkedserver example with DSN-less connection */

/* This example also depends on the sample files Testdata.dbc

Modify your code accordingly for differences in location or DBC name */

sp_addlinkedserver 'VFP Testdata Database With No DSN',

'',

'MSDASQL',

NULL,

NULL,

'Driver={Microsoft Visual FoxPro
Driver};UID=;PWD=;SourceDB=e:\VFP8junk\Testdata.dbc;SourceType=DBC;Exclusive=No;BackgroundFetch=Yes;Collate=Mac
hine;'

go

SELECT *

FROM OPENQUERY([VFP Testdata Database With No DSN], 'select * from customer where country = "USA" order by
country')

go

```

参考

有关设置和使用分布式查询的详细信息，请参阅 `sp_addlinkedserver` SQL 7.0 联机丛书中的 "查看"、"OpenQuery"、"OpenRowset" 和 "相关主题"。

若要了解有关 FoxPro 和文件的详细信息 `.dbf` `.dbc`，请参阅 FoxPro 产品文档。

使用 VFP MSComm32 控件传输和接收二进制数据

2020/10/22 •

本文介绍如何使用 MSComm32 控件通过不带调制解调器的 RS-232 电缆接收和传输二进制数据。

原始产品版本: Visual FoxPro

原始 KB 数: 154741

摘要

本文介绍了一些使用 Visual FoxPro 3.0 通信 (MSComm) 控件用于通过 RS-232 (电缆接收和传输二进制数据的方法, 而无需使用调制解调器)。

更多信息

通信控制 MSCOMM32.OCX, 通过允许通过串行端口传输和接收仅允许传输缓冲区中的字符串的数据, 从而为您的应用程序提供串行通信。此功能限制您仅传输基于文本的文件。

本文演示如何使用 MSComm 控件来传输和接收使用 RS-232 电缆的二进制数据。(本文中的技术也适用于调制解调器。) 必须将二进制数据一次转换为1个字节, 然后再进行传输。收到时, 必须一次将数据从一个字符转换回二进制数据1字节。

在启用 DBCS 的操作系统 (运行使用双字节字符集之一的操作系统软件时), 如果其中一个二进制值与 DBCS 前导字符相匹配, 则二进制数据将被损坏。MSComm 控件将此字节和以下字节解释为一个双字节字符, 并只为等效的 ASCII 字符返回1个字节。若要解决此问题, 需要将1个字节转换为三个字符的 ASCII 大小, 以保留前导字符。

示例代码

以下代码演示了此过程。

WARNING

使用本文中提供的示例代码的风险由您自己承担。Microsoft 以 "原样" 提供此示例代码, 而无需任何明示或暗示的担保, 包括但不限于对适销性和/或适用于特定用途的暗示担保。

```
* Transmitter Code.

* INIT event of Comm OLE control
* 28800 baud, no parity, 8 data, and 1 stop bit.
* In RS-232, maximum speed of 28800 baud can be used

This.Settings = "28800,N,8,1"
This.InputLen = 1
This.CommPort = 1
This.PortOpen = .T.

* ONCOMM event of Comm OLE Control
* The following code is needed to make sure that next set of characters
* can be transmitted (CommEvent = 2 is triggered from the receiver side)

IF This.CommEvent = 2
  This.input
  IF gnTop <= gnEnd
    gcString = FREAD(gnFileHandle, 1) && Store to memory
    q=asc(gcstring)
    * change ASCII to character (size of 3) to preserve the lead char
```

```

thisform.olecontrol1.output = str(q,3)
gnTop = gnTop + 1
ENDIF
ENDIF

* INIT event of form
PUBLIC gnFileHandle
PUBLIC gnEnd
PUBLIC gnTop
PUBLIC q
* You should replace 'c:\sample.hlp' with your own binary file
STORE FOPEN('c:\sample.hlp') TO gnFileHandle && Open the file
STORE FSEEK(gnFileHandle, 0, 2) TO gnEnd && Move pointer to EOF
STORE FSEEK(gnFileHandle, 0) TO gnTop && Move pointer to BOF
gntop=1
q=""
-----
Property of OleControl1

RThreshold = 1 * triggers when at least one char is on the buffer
SThreshold = 3
-----

* Receiver Code.

* INIT event of OleControl1
* 28800 baud, no parity, 8 data, and 1 stop bit.

This.Settings = "28800,N,8,1"
This.InputLen = 3
This.CommPort = 1
This.PortOpen = .T.

* OnComm event

IF This.CommEvent = 2 AND This.InBufferCount > 0
qq=CHR(VAL(This.Input))
=FWRITE(gnFileHandle,qq)
this.output = CHR(26)
ENDIF
* INIT event of form
PUBLIC gnFileHandle
* You should replace 'c:\sample.hlp' with your own file name
IF FILE('c:\sample.hlp') && Does file exist?
gnErrFile = FOPEN('c:\sample.hlp',12) && If so, open read-write
ELSE
gnErrFile = FCREATE('c:\sample.hlp') && If not, create it
ENDIF
= FCLOSE(gnErrFile) && Close the file
STORE FOPEN('c:\sample.hlp',1) TO gnFileHandle && Open the file
* CLICK event of button
*This tells the transmitting side to start sending the file
Thisform.Olecontrol1.output = CHR(26)
-----
Property of OleControl1
RThreshold = 3
-----

```

当输入缓冲区中存在等待的字符时，事件驱动技术将生成 OnComm 事件。此外，该 `CommEvent` 属性将包含数字2。对于要触发的 OnComm 事件，您必须将该 `Rthreshold` 属性设置为非零值 (其默认)。该属性最常见的设置 `RThreshold` 为 1，这意味着，如果在输入缓冲区中至少有一个字符在等待，则会触发 OnComm 事件。在本文中，3 `RThreshold` 由于一次发送了三个字符，因此将3用作属性的值。

如何在 Visual FoxPro 中解决可疑损坏问题

2020/10/22 •

本文提供了开始解决在 Visual FoxPro 中遇到的数据损坏问题的基本步骤。

原始产品版本: Visual FoxPro

原始 KB 数: 193952

摘要

使用 Microsoft Visual FoxPro 时,有时可能会遇到问题。这些问题的原因并不总是显而易见。在这种情况下,您需要一个常规方法或一种试用和错误故障排除技术的过程。"详细信息"一节中的步骤提供了疑难解答过程的开始步骤。这些步骤也适用于以在客户端计算机上安装的 Visual FoxPro 编写的分布式应用程序 .exe 文件。

第一步是确保您在 Visual FoxPro 产品中安装了错误的无错误。在我们的经验中,如果您在与早期版本相同的文件夹中安装最新版本,则安装不会正确完成或不可靠。这是因为所有文件可能无法正确更新或安装。

更多信息

1. 如果在本地计算机上安装了 Visual FoxPro, 请按照以下步骤操作:

- a. 备份文件。保存要保留的计算机上的任何文件;尤其是数据库 (.dbc/.dct/.dcx) 文件和表 (.dbf/.fpt/.cdx) 文件以及其他源文件(例如 .pjx, .pjt, .vcx, .vct, .scx, .sct, .mnx, .mnt, .frx, .frt 和 fpw 文件)。
- b. 完全删除 Visual FoxPro, 然后删除安装它的目录。

NOTE

如果您在步骤3中,则无需执行此步骤。您需要安装到全新的子目录中。

- c. 关闭所有打开或运行的应用程序。使用 CTRL + ALT + DELETE 打开系统任务管理器,使您可以查看当前正在运行的应用程序。

NOTE

请勿关闭资源管理器,因为它是桌面用户界面。

- d. 使用 "磁盘扫描程序" 或第三方磁盘修复工具执行操作系统磁盘错误检查。这些工具会检查并尝试修复硬盘上的任何丢失的群集。然后, **Defragmenter** 在 WINDOWS XP) 或其他工具中运行磁盘实用程序 (以优化磁盘。

2. 再次关闭所有打开或运行的应用程序, (请参阅 step 1c) 。

3. 将 Visual FoxPro 安装到新的或不同的目录位置。

成功的产品安装意味着您在安装过程中不会收到任何错误。安装过程中的错误指示不可靠的 Visual FoxPro 安装。如果发生错误:

- a. 请注意安装过程中出现的任何错误。
- b. 在安全模式或带网络的安全模式下启动计算机。此外,暂时禁用任何防病毒软件。
- c. 再次从步骤1开始。

NOTE

如果这是第二次执行此过程, 请尝试使用自定义、最小或便携式电脑安装, 而不是完整安装。

4. 如果在网络服务器上安装了 Visual FoxPro:

- 您必须在 Visual FoxPro 中安装无错误的错误。
- 通常情况下, 在本地区域网络 (LAN) 或服务器运行时, 无法执行磁盘扫描或其他磁盘完整性检查。
- 删除并重新安装 Visual FoxPro, 直到未收到安装错误。

5. 消除所有重复的源代码和项目文件。这包括网络驱动器和本地驱动器。

重复的文件会导致多个问题。请确保保留源文件或项目文件的正确版本, 并且仅保留正确的版本。

6. 如果项目包含数据库/DBC, 请验证数据库。

NOTE

有关详细信息, 请参阅 Visual FoxPro 帮助文件中的 "打开数据库" 命令。

使用 "开放式数据库 独占验证" 命令。

错误表示表 (.dbf) 或索引 (.cdx) 损坏。

7. 清理项目。

打开您的项目, 然后从 "项目" 菜单中选择 "清理项目"。

错误表示项目 (.pjx) 文件损坏。

8. 作为一种使用其他文件进行测试的方法, 请使用文件 (.dbf), 将 .dbf 结构复制到新文件中, 并追加原始表中的记录, 并对这些文件进行重命名。如果需要, 此方法需要创建新的索引 () 文件。

NOTE

下面的步骤使用名为 "Customer" 的 (.dbf) 文件的示例表。

数据库、项目、屏幕、菜单、报表和标签文件实际上是表 (的 .dbf) 扩展名不是 .dbf 的文件。

如果您的文件已在 Visual FoxPro 交互或设计环境中打开, 请将其关闭。

在 Visual FoxPro 命令窗口中发出以下命令:

```
USE customer.dbf && Must specify extension if other than .dbf.
? RECCOUNT() && Note the number of records in the table.
COPY STRUCTURE TO newfile.dbf
USE newfile.dbf && Empty file structure without records.
APPEND FROM customer.dbf && Bring in records from original table.
? RECCOUNT() && Compare number of records with original file.
RENAME customer.dbf TO oldcust.dbf
RENAME newfile.dbf TO customer.dbf
```

NOTE

与原始文件相比, 新文件中的记录数不同。这是在原始文件中存在文件损坏的指示器。

9. 在多用户环境中或使用其他操作系统时, 在不同计算机或不同的计算机环境 (如单个用户环境) 中进行测试。

10. 如果损坏, FoxPro 资源文件可能会引起问题。

测试资源文件是否是一个因素的一种方法是替换文件。以下两种方法可以确定 FoxPro 资源文件的名称和位置:

- a. 您可以使用以下 FoxPro 命令确定当前活动资源文件的位置: `?SYS(2005)`
- b. 默认情况下, 可以在 FoxPro 配置文件中指定 FoxPro 资源文件(名为 fpw)。如果配置文件所指定的资源文件不存在, 则 FoxPro 将创建一个。

- a. 使用以下 FoxPro 命令确定配置文件的位置: `?SYS(2019)`
- b. 配置文件是 ASCII 文本文件, 可以使用任何以 ASCII 文本格式保存它的编辑器进行编辑。编辑当前活动的 FoxPro 配置文件的最简单方法是使用以下命令: `MODIFY FILE SYS(2019)`

- c. 配置文件可能包含一行文本, 用于指定资源文件的位置和名称, 如下所示:

```
RESOURCE=<path>\foxuser.dbf
```

如果是这样, 通过在行的开头放置一个星号 (*) 注释掉下划线。

- d. 添加新行, 如下所示:

```
RESOURCE=<path>\newjunk.dbf && Any name of your choice.
```

- e. 重新启动 FoxPro 以使新设置生效。

11. 创建一个包含上述步骤的结果的日志文件, 以供将来参考。

参考

有关 Visual FoxPro 的设置要求的详细信息, 请参阅位于 Visual FoxPro 安装目录中的 Readme.txt Windows 帮助文件或原始安装 cd-rom 上的 ".hlp"。

Visual FoxPro 帮助;搜索条件: "安装 Visual FoxPro"、"测试和调试应用程序"。

对窗体控件使用 KeyPress 事件

2020/10/22 •

本文介绍如何对窗体控件使用 KeyPress 事件。

原始产品版本: Visual FoxPro

原始 KB 数: 140296

摘要

许多开发人员使用关键标签在程序中分配要在按特定键时执行的程序或过程, (通常是) 的功能键。但是, 如果键标签分配特定于表单中的控件, 则更好的方法是使用该控件的 KeyPress 事件。

更多信息

KeyPress 事件适用于以下表单控件:

CheckBox、ComboBox、命令按钮、EditBox、Form、ListBox、选项按钮、微调框、TextBox

可以使用此事件来检查是否按下了某个特定键。使用 KeyPress 事件根据函数键执行操作时, 应使用函数键的值, 如下表所示:

Key	Alone	Shift	Ctrl	Alt

F1	28	84	94	104
F2	-1	85	95	105
F3	-2	86	96	106
F4	-3	87	97	107
F5	-4	88	98	108
F6	-5	89	99	109
F7	-6	90	100	110
F8	-7	91	101	111
F9	-8	92	102	112
F10	-9	93	103	113
F11	133	135	137	139
F12	134	136	138	140

有关键值的详细信息, 请参阅 `INKEY()` Visual FoxPro 帮助文件中的函数。

分步示例

下面的步骤演示如何使用 KeyPress 事件。

1. 创建一个窗体, 并在其上放置一个文本框。
2. 在 `ControlSource` 文本框的属性中, 键入 `mykey` (或您可能首选) 的任何其他变量。
3. 将以下代码放在文本框的 KeyPress 事件中:

```

LPARAMETERS nKeyCode, nShiftAltCtrl
*: The previous line of code is already contained in the KeyPress event
*: by default.
DO CASE

CASE nKeyCode = 28
WAIT WINDOW "The F1 Key has been pressed" NOWAIT
mykey = "F1 Key"
ThisForm.Refresh

CASE nKeyCode = -4
WAIT WINDOW "The F5 Key has been pressed" NOWAIT
mykey = "F5 Key"
ThisForm.Refresh

ENDCASE

```

4. 将以下代码放在窗体的 Load 事件中：

```

CLEAR MACROS
SET HELP OFF
*: This will disable FoxPro's default function key assignments.

```

5. 将以下代码放在表单的销毁事件中：

```

RESTORE MACROS
SET HELP ON
*: This will restore FoxPro's default function key assignments.

```

6. 将命令按钮放置在窗体上，并将以下代码放在其 click 事件中：

```

ThisForm.Release

```

7. 保存并运行该窗体。如果按 F1 或 F5 键，则应显示相应的等待窗口消息，并且应更新文本框。

使用 FoxPro 检查是否存在目录

2020/10/22 •

本文介绍如何使用 FoxPro 检查是否存在目录。

原始产品版本: Visual FoxPro

原始 KB 数: 94339

摘要

有两种方法可检查驱动器上的目录是否存在:

- `FILE()` 函数可用于检查本地驱动器上的文件是否存在。`FILE()` 仅适用于工作组下的网络驱动器。

NOTE

对 windows `FILE()` 3.11 下的 MS-DOS 运行 Windows foxpro 或 foxpro 时, 使用函数所述的方法不起作用。此方法仅在以下情况下使用: 在高于 3.0 的 MS-DOS 版本下直接运行时, 或者在 Windows NT 下为 ms-dos 或 Windows NT 运行 foxpro 时运行 foxpro。

- `ADIR()` 和 `ASCAN()` 函数可用于共同检查本地或网络驱动器上的文件是否存在。

更多信息

`FILE()` 尝试在磁盘上查找文件或目录, 并返回 .T. 或 .F. 基于搜索结果。指定要作为字符字符串参数搜索的文件名。若要测试某个目录是否存在, 请将 `NUL` 其指定为文件名。例如:

```
?FILE('c:\temp\NUL')
```

上面的命令检查是否存在该 `C:\FOXPRO2` 目录。

NOTE

版本 3.0 之前的 MS-DOS 版本不支持使用 `NUL` 的这种用法。

`ADIR()` 将文件信息放在数组中。生成的数组包含五列, 分别包含文件名、大小、日期、时间和属性。`ADIR()` 最大使用三个参数:

- `<expC1>` 指示数组名称。
- `<expC2>` 是可选的, 可用于指定模板文件名, 如 `C:\FOX*` 或 `*.DBF`。只有与模板匹配的文件名才会放置在数组中。
- `<expC3>` 可以包含字母 D、H、S 或 V, 分别将结果限制为目录名称、隐藏文件、系统文件或卷名称。

`ASCAN()` 然后, 可以使用该函数确定是否在数组中放置了匹配的目录条目。例如:

```
?ADIR(dirarray, 'FOXPRO2', 'D')
```

上面的命令创建一个名为的数组 `dirarray`, 并为该目录创建一行 `FOXPRO2`。


```
?ASCAN(dirarray,'FOXPRO2')
```

上面的命令返回.T。或.F。如果 `FOXPRO2` 数组中包含, 则为。

参考

有关、和的详细信息, 请 `FILE()` `ADIR()` `ASCAN()` 参阅 FoxPro 版本 2.0 "命令 & 函数" manual or FoxPro Version 2.5 "Language Reference" 手册。

在 Visual FoxPro 中使用 null 值

2020/10/22 •

本文介绍如何在 Visual FoxPro 命令和函数中使用 null 数据值。

原始产品版本: Visual FoxPro

原始 KB 数: 130307

摘要

Visual FoxPro 版本3.0 支持 null 数据值。版本3.0 之前的 FoxPro 版本不直接支持 null 数据值。本文介绍 Visual FoxPro 中的常规规则, 以处理 null 值, (`.NULL.` 在传递给 Visual FoxPro 命令或函数时表示为) 。

更多信息

`.NULL.` 当缺少值、不相关或未知值时, 请使用 null 值 () 值。在早期版本的 FoxPro 中, 未知或缺失的值由空格或零表示, 这些值可能已被误解。使用 Visual FoxPro, 您现在可以在字段中存储 null 值。

请务必注意, (`.NULL.`) 在 Visual FoxPro 中的 null 值与空、空或零值不相同。Null 表示缺少值, 因此 null 永远不等于、大于或小于另一个值, null 或非 null。Visual FoxPro 对 null 值的支持符合 ANSI 标准, 并影响使用值和表达式的产品的任何区域。

Null 值的通用规则

以下是传递给 Visual FoxPro 命令或函数的 null 值的一般规则:

- 命令在传递空值时生成错误。
- 因此, 接受 null 值的函数将返回 null 值 `.NULL.` 。
- 如果函数的值为 null, 则需要数值的函数将生成错误。
- `ISBLANK()`、`ISDIGIT()`、`ISLOWER()`、`ISUPPER()`、`ISALPHA()` 和 `EMPTY()` 每个在传递 null 值时 () 的返回 false。
- `ISNULL()` 传递 null 值时, 返回 true () 。
- `INSERT` 通过的命令 SQL 和 `SELECT` sql 过程中的 NULL 值为 null, 并且不是 null 子句。
- SQL 聚合函数(如 `MAX()`、`MIN()` 和) `SUM()` 将忽略聚合中的所有 null 值。
- `.NULL.` 如果所有提供的值均为 null 值, 则 Visual FoxPro 聚合函数将传播, 否则, 将忽略任何 null 值。

本文的其余部分提供了有关这些通用规则的详细信息和示例。

命令在传递空值时生成错误

Visual FoxPro 命令是一个可导致操作的语句。、和的命令 `USE` 示例 `BROWSE` `DELETE` 。例如, `USE` 该命令为此代码返回一个错误:

```
STORE .NULL. TO nWorkArea
USE mytable IN (nWorkArea)
```

命令的 IN 子句 `USE` 需要一个数值或 alpha 值, 当传递的 `.NULL.` 错误 **表编号无效** 时, 将生成该编号。

`NVL()` 函数可用于从计算或运算中删除 null 值, 其中 null 值不受支持或不相关。

```
STORE .NULL. TO nWorkArea
USE mytable IN NVL(nWorkArea,0)
```

这将在第一个可用的 workarea 中打开 mytable。有关函数的详细信息, 请参阅帮助文件或 Visual FoxPro 文档

`NVL()`。

接受 null 值的函数将返回。不适。因此

直观的 FoxPro 函数是执行特定任务并采用零个或多个参数的例程。函数的示例包括 `ISBLANK()`、`UPPER()` 和 `SUBSTR()`。大多数可视 FoxPro 函数允许将 null 值作为参数传递, 而不生成错误, 但是 `.NULL.` 从函数返回。换言之, 在将 null 值传递给函数时, 结果始终为 null。这也是在数学等式中处理 null 值的方式。例如, 向 500 添加的 null 值等于 null, null 值乘以零等于 null (不是零)。

下面的代码示例返回 `.NULL.`：

```
cLastName = "Johnson"
nBegin = 5
nExtract = .NULL.
?SUBSTR(cLastName,nBegin,nExtract)
```

此规则的例外是、、、`ISBLANK()` `ISDIGIT()` `ISLOWER()` `ISUPPER()` `ISALPHA()` 和 `EMPTY()` 函数-每个都返回一个 `.F.` 值。`ISNULL()` 函数返回一个 `.T.` 值。

使用 new 子句插入 SQL 并选择 SQL 过程 Null 值

(的两个新子句为 NULL 且不为 NULL) 在 `INSERT` 和 SQL 命令中处理 null `SELECT`。例如, 若要查找 cLastName 不为 null 的表中的所有记录, 请使用以下命令：

```
SELECT cLastName FROM mytable WHERE cLastName IS NOT NULL
```

若要查找 null 值, 请使用为 NULL 子句。

SQL 聚合函数忽略空值

聚合函数是一种函数, 用于对组 (聚合) 值执行数值运算, 如加法、最小值、最大值或平均值。聚合函数的示例包括 `MAX()`、`MIN()` 和 `SUM()`。

`SELECT` 例如, SQL 命令可以使用聚合函数从表中检索数字值。例如, 以下 `SELECT` 命令返回名为 ":" 的字段中的最大值 `nYTDSales`。

```
SELECT MAX(nYTDSales) from mytable
```

对包含值的字段执行的任何 SQL 聚合函数 `.NULL.` 都会忽略这些 `.NULL.` 值, 返回的结果会将空值视为不是聚合的一部分, 而不是它们。

有关 `.NULL.` 上面所述的值和函数的详细信息, 请在 Visual FoxPro 帮助文件中搜索主题。

将 "追加常规" 命令用于 Imager 文件

2020/10/22 •

本文介绍如何将 "追加常规" 命令用于 Imager 文件。

原始产品版本: Visual FoxPro

原始 KB 数: 152370

摘要

本文介绍如何使用 `APPEND GENERAL` 命令将 PICT 文件放在 Visual FoxPro 常规字段中。它还提供了一些有关可能发生的问题以及如何纠正这些问题的信息。

`APPEND GENERAL` 在 Visual FoxPro 中追加 Imager 文件的命令的一般语法为:

```
APPEND GENERAL <generalfield> FROM <filename> ;  
CLASS "Microsoft Imager 2.0 Picture"
```

NOTE

此命令应全部在一行上。由于空间限制, 它在两行上显示。

若要确定在命令上为 CLASS 子句输入的内容 `APPEND GENERAL`, 请在 Visual FoxPro 中修改常规字段。然后从 "编辑" 菜单中选择 "插入对象" 命令。列出的项目是将用于该命令的类名称 `APPEND GENERAL`。

更多信息

Visual FoxPro 附带有的一些 PICT 文件。它们位于 "Microsoft Visual FoxPro: Samples: Graphics" 文件夹中。这些图形包含在自解压存档 (海平面) 文件中。若要执行以下步骤, 请双击 Picts 以解压缩 PICT 文件。将它们存储在指定的 "Picts 文件夹" 的默认文件夹中。

分步示例

若要使用 `APPEND GENERAL` 命令将 PICT 文件放置在 Visual FoxPro 常规字段中, 需要执行以下步骤:

1. 在 Microsoft Imager 中打开 PICT 文件。Imager 还可以) 文件中打开 TIFF (Tag 图像文件格式。打开 "示例: 图形: Pict 文件夹: Fox: 添加"。PICT 文件。
2. 从 "文件" 菜单中选择 "另存为" 以保存图像文件。在 "另存为" 对话框中, 将文件类型更改为 "Macintosh PICT"。选择 "另存为" 对话框中的选项以更改 PICT 文件的属性。对于本文的目的, 默认值是不合适的。将文件保存在 Imager 中会将创建者更改为 MIMG, 以便 OLE 能够使用它。
3. 保存添加。PICT 文件为 `ADDImager`。关闭 Imager 中的 PICT 文件。
4. 在 Visual FoxPro 中, 创建或打开一个包含 "常规" 字段的表。向表中追加一个空白记录, 如 "追加空"。在 Visual FoxPro 命令窗口中, 发出以下命令:

```
APPEND GENERAL olefield FROM ;  
HOME()+"Samples:Graphics:Fox:ADDImager.PICT" ;  
CLASS "Microsoft Imager 2.0 Picture"
```

NOTE

上面的命令应全部在一行上。由于空间限制，此处将此处显示为三行。您的常规字段名称和 PICT 文件的路径可能会有所不同。

此 PICT 文件应追加无问题。不过，它很小。比 Visual FoxPro 附带的样本更大的 PICT 文件可能需要更多内存才能分配给 Imager，然后才能成功追加。默认情况下，Imager 已向其分配 4096K。将此增加到 8192 K 将允许将较大的 PICT 文件以 "追加常规" 命令插入到 "常规" 字段中，或者通过浏览界面并在 "常规" 字段打开时从 "编辑" 菜单中选择 "插入对象" 命令 (MODIFY olefield)。

可以使用以下程序自动将 PICT 文件的创建者更改为 Imager，并将 PICTS 追加到常规字段中。该程序假定在 Microsoft Visual FoxPro: Samples: Graphics folder 中找到 Picts 文件，以下是大纲 PICT 文件的文件夹路径：

```
<foxpro folder>:Samples:Graphics:Picts folder:Outline
```

只有六个大纲 PICT 文件。如果涉及大量文件，此过程可能需要很长时间。

```
SET LIBRARY TO HOME()+"FoxTools.cfm" ADDITIVE
DIMENSION farray(1,1)
CLEAR

y=HOME()+"Samples:Graphics:Picts folder:Outline:*.PICT"
x=ADIR(farray,"&y")? "Changing Creators of the Pict files to Imager."
?
FOR i=1 TO x
IF FxSetType(SYS(2027,HOME()+ ;
"Samples:Graphics:Picts folder:Outline:"+farray(i,1)), "PICT", "MIMG")=0
WAIT farray(i,1)+" changed successfully" TIMEOUT 2
ELSE
WAIT farray(i,1)+" not changed successfully" TIMEOUT 2
ENDIF
ENDFOR

SELECT 0
CREATE TABLE OutlinePicts (descript C(32), olefield G)?
?
?
? "Appending in the Pict files"
?
FOR i=1 TO x
APPEND BLANK
REPLACE OutlinePicts.descript WITH farray(i,1)
y=SYS(2027,HOME()+"Samples:Graphics:Picts folder:Outline:"+farray(i,1))
APPEND GENERAL olefield FROM (y) CLASS "Microsoft Imager 2.0 Picture"
WAIT farray(i,1)+" appended" TIMEOUT 2
ENDFOR

BROWSE NOWAIT
SET LIBRARY TO
```

在 Visual FoxPro 中使用 DateTimePicker 控件

2020/10/22 •

本文介绍如何在 Visual FoxPro 中使用 DateTimePicker 控件。

原始产品版本: Visual FoxPro

原始 KB 数: 259646

摘要

DTPicker.exe 是演示如何使用 ActiveX DateTimePicker 控件的示例。

详细信息

以下文件可从 Microsoft 下载中心下载:

[立即下载 DTPicker.exe 程序包](#)。发布日期: Apr-26-2000

有关如何下载 Microsoft 支持文件的详细信息, 请参阅 [如何从联机服务获取 Microsoft 支持文件](#)。

Microsoft 已对此文件进行病毒扫描。Microsoft 使用了最新的病毒检测软件, 该软件在文件发布之日起可用。文件存储在安全增强的服务器上, 可帮助防止对文件进行任何未经授权的更改。

DTPicker.exe 文件包含以下文件:

文件名	SIZE
ReadMe.txt	3560
DTPicker	4630
DTPicker	19897

关于 DateTimePicker 控件

DateTimePicker 控件使您能够提供允许轻松进行日期或时间选择的格式化日期/时间字段。此外, 用户可以从类似于 MonthView 控件的下拉日历界面中选择日期。

DateTimePicker 控件在两种模式下运行:

- 下拉日历模式 (默认) -使用户能够显示可用于选择日期的下拉日历。
- 时间格式模式-使用户能够在日期显示中选择一个字段 (例如, 月、日、年等等), 然后按控件右侧的向上或向下箭头以设置其值。

您可以使用键盘或鼠标导航到控件。下拉日历包含两个按钮, 可让您在视图中滚动月份。

NOTE

DateTimePicker 控件是 Mscomctl2 文件中找到的一组 ActiveX 控件的一部分。若要在应用程序中使用 DateTimePicker 控件, 必须将 Mscomctl2 文件添加到项目中。分发应用程序时, 请在用户的 Microsoft Windows System 或 System32 目录中安装 Mscomctl2 文件。

关于此示例

本示例提供一个包含 DateTimePicker 控件的窗体以及其他控件, 这些控件使您能够操作控件的各种属性并查看结果。许多控件都设置为只读, 因为它们代表的 DateTimePicker 控件中的属性是只写的。您可以从 `Object.Value` 控件的属性或与 datetime 值的各个部分对应的属性(年、月、日、小时、分钟、秒等)访问当前日期/时间值。

通过 DateTimePicker 控件, 您可以指定日期/时间的有效范围以及控件中显示的 datetime 值的格式。可用的格式为长日期 (0)、短日期 (1)、时间 (2) 和自定义 (3)。如果指定了自定义格式, 则可以使用 DateTimePicker 控件支持的格式字符来自定义满足您的需求的格式。

NOTE

`CustomFormat` 属性区分大小写。虽然 M 和 MM 格式引用月份, 但小写值 m 和 mm 表示的是分钟数。您还需要注意的是 h、hh、H 和 H 指的是小时;小写值 h 和 hh 是指以12小时制格式的小时数, 而大写的 H 和 HH 指的是24小时格式的小时数。

如前所述, DateTimePicker 可以(下拉日历或时间格式)中的一种模式运行。此属性由属性控制 `UpDown`。当 `UpDown` 属性设置为 `True` (), 控件处于时间格式模式。通过控件最右侧的微调按钮(如微调按钮)的外观, 这是显而易见的。在此模式下, 用户可以选择日期时间值的一部分(月、日、年、小时、分钟等)并通过这些值旋转, 直到达到所需的日期/时间。当该 `UpDown` 属性设置为 `False` 时 () 该控件处于下拉日历模式。在控件最右侧的下拉按钮的外观中, 这是显而易见的。在此模式下, datetime 值的时间部分将不会更改。

NOTE

当 DateTimePicker 控件的格式设置为 Time (2) 时, 该 `UpDown` 属性将被忽略, 并且该控件处于时间格式模式。

DateTimePicker 控件可以在控件中启用 checkbox。清除此复选框(取消选中)后, 从控件返回的日期和时间值为。不适. 值。

参考

有关 DateTimePicker 控件的详细信息, 请参阅 Windows 控件帮助文件 CMCTL298 中的主题。CHM, 位于已安装 MSDN 库的位置。

有关如何向项目中添加 ActiveX 控件的详细信息, 请参阅程序员指南中的 "将控件添加到项目"。

使用 "将打印机设置为名称" 命令在 Visual FoxPro 中打印到指定打印机

2020/10/22 •

本文介绍如何使用 "将打印机设置为名称" 命令打印到指定打印机。

原始产品版本: Visual FoxPro

原始 KB 数: 162798

摘要

为了指定将 Visual FoxPro 报告发送到特定打印机, 习惯于更改 Windows 默认打印机或使用 `PROMPT` 命令上的子句, `REPORT` 以允许用户选择打印机。但是, 可以更改所需的目标打印机, 而无需使用这些技术中的任何一种。本文介绍使用 "将打印机设置为名称 [打印机名称]" 来执行此操作的一些方法。

更多信息

"将打印机设置为名称" 命令允许将 Visual FoxPro 中当前选定的打印机设置为安装在 Windows 下的任何打印机。

已安装的打印机列表可以通过函数读入阵列 `APRINTERS()`, 也可以通过对话框返回单个打印机名称 `GETPRINTER()`

。

NOTE

若要执行此过程, 必须使用在 "打印设置" 对话框内的 "名称" 下拉框中选择的默认打印机保存报告表单。此对话框从 "文件" 菜单下的 "页面设置" 对话框中调用。

示例 1

下面的程序显示了 `GETPRINTER()` 允许从已安装打印机中选择打印机的对话框。如果在 `GetPrinter` 对话框中选择了 "确定", 则会向所选打印机发送名为 Test 的报告。如果在 "GetPrinter" 对话框中选择了 "取消" 按钮, 则不会打印该报表。

```
* Beginning of program for Example 1
* *****
* This program assumes a report called Test which has been
* saved with the default printer in the Print Setup dialog.
*
LOCAL lcPrinter

lcPrinter = ''
lcPrinter = GETPRINTER()

IF !EMPTY(lcPrinter)
SET PRINTER TO NAME (lcPrinter)
REPORT FORM Test TO PRINTER NOCONSOLE
ELSE
WAIT WINDOW 'Print Job Cancelled' TIME 1
ENDIF
*
* End of program for Example 1
```


NOTE

在 Visual FoxPro 3.0 和 3.0 b 下, 该 `GETPRINTER()` 对话框是不同的, 对话框允许更改纵向/横向设置, 以及其他。打印报表时, 将忽略这些情况。

示例 2

下面的程序显示一个自定义窗体, 其中包含一个使用 `APRINTERS` 创建的数组, 其中包含一个由 (# A1 创建的数组。如果选择 "打印报告" 命令按钮, 则会向所选打印机发送名为 Test 的报告。如果选择 "取消" 按钮, 则不会打印报告。

```
* Beginning of program for Example 2
*****
* This program assumes a report called Test which has been
* saved with the default printer in the Print Setup dialog.
*
PUBLIC oform
oform=CREATEOBJECT("form1")
oform.SHOW

DEFINE CLASS form1 AS FORM
    AutoCenter = .T.
    Height = 158
    Width = 327
    Caption = "Printer Dialog"
    Name = "Form1"
    DIMENSION aprintarray[1]

    ADD OBJECT combo1 AS COMBOBOX WITH ;
        ROWSOURCETYPE = 5, ;
        ROWSOURCE = "thisform.aPrintArray", ;
        HEIGHT = 25, ;
        LEFT = 24, ;
        STYLE = 2, ;
        TOP = 48, ;
        WIDTH = 276, ;
        NAME = "Combo1"

    ADD OBJECT command1 AS COMMANDBUTTON WITH ;
        TOP = 108, ;
        LEFT = 48, ;
        HEIGHT = 27, ;
        WIDTH = 84, ;
        CAPTION = "Print Report", ;
        DEFAULT = .T., ;
        NAME = "Command1"

    ADD OBJECT command2 AS COMMANDBUTTON WITH ;
        TOP = 108, ;
        LEFT = 180, ;
        HEIGHT = 25, ;
        WIDTH = 84, ;
        CANCEL = .T., ;
        CAPTION = "Cancel", ;
        NAME = "Command2"

    PROCEDURE combo1.INIT
        LOCAL lnI
        FOR lnI = 1 TO APRINTERS(THISFORM.aprintarray)
            * Note below that you are adding a leading space. This
            * prevents a network printer in Windows NT from appearing
            * disabled in the combo due the leading "\"
            THISFORM.aprintarray[lnI,1] = SPACE(1) + ;
            THISFORM.aprintarray[lnI,1]
        ENDFOR
        * Set initial value of combo
        THIS.Refresh()
```

```

* This sets the combo initial value of the dropdown to the
* default printer - This will not work in 3.0/3.0b since
* SET('PRINTER', 2) is not available
FOR EACH a_element IN THISFORM.aprintarray
IF UPPER(SET('PRINTER',2))$UPPER(a_element)
THIS.Value = a_element
ENDIF
ENDFOR
ELSE
* If in 3.0/3.0b, set to first element in list.
THIS.Value = THISFORM.aprintarray[1]
ENDIF
ENDPROC

PROCEDURE command1.CLICK
SET PRINTER TO NAME (ALLTRIM(THISFORM.combo1.VALUE))
REPORT FORM test TO PRINTER NOCONSOLE
RELEASE THISFORM
ENDPROC

PROCEDURE command2.CLICK
RELEASE THISFORM
ENDPROC

ENDDDEFINE
*
* End of program for Example 2

```

参考

- [Visual FoxPro 5.0 帮助](#)
- [Visual FoxPro 6.0 帮助](#)

在 Visual FoxPro 中使用 SQL-SELECT 语句中的通配符

2020/10/22 •

本文介绍如何在 SQL-SELECT 语句中使用通配符以将多功能性添加到此语句中。

原始产品版本: Visual FoxPro

原始 KB 数: 98434

摘要

若要扩大结构化查询语言的选择, (SQL-SELECT) 语句, 两个通配符, 可以使用百分号 (%) 和下划线 (_)。

百分比符号类似于星号 (* MS-DOS 中使用) 通配符字符。百分号符号允许替换字段中的一个或多个字符。

下划线类似于 MS-DOS 通配符问号字符。下划线允许替换表达式中的单个字符。下面提供了三个示例, 演示如何在 SQL-SELECT 语句中使用这些字符。这些示例使用在 Home (2) + data 文件夹中包含在 Microsoft Visual FoxPro 中的 "Customer.dbf" 文件。

更多信息

以下 SELECT-SQL 语句返回 Customer 表中的所有记录, 其中包含 "W" 作为字段中的第一个字符

Regionabbreviation。SELECT-SQL 语句在浏览窗口中列出来自 "WY" 和 "WA" 的记录。

```
SELECT * from customer WHERE Region like "W_"
```

以下 SELECT-SQL 语句列出了所有包含 "P" 或 "C" 的记录作为字段中的最后一个字符 Regionabbreviation。输出包括来自 "SP" 和 "BC" 的记录。

```
SELECT * from customer WHERE Region like "_P" OR region like "_C"
```

下面的 SELECT-SQL 语句使用百分号和下划线返回其 Cust_ID 以 "G" 开头且具有以 "P" 结尾的字段的所有记录

Regionabbreviation。

NOTE

百分号符号使任何字符串的字符在 "G" 之后。相反, 下划线只允许替换单个字符。

```
SELECT * FROM Customer WHERE cust_ID LIKE "G%" AND region LIKE "_P"
```

若要执行类似于 \$ 函数所执行的字符串搜索, 请使用如下语句:

```
SELECT * from customer WHERE company like "%M%"
```

此 SELECT-SQL 语句查找名称包含 "M" 的所有公司。

参考

有关详细信息, 请参阅 Visual FoxPro 帮助文件。搜索 "SELECT-SQL"。