返回总目录

OLEServerBusyTimeout 属性
OLESetData 事件
ON BAR 命令
ON ERROR 命令
ON ESCAPE 命令
ON EXIT BAR 命令
ON EXIT MENU 命令
ON EXIT PAD 命令
ON EXIT POPUP 命令
ON KEY 命令
ON KEY = 命令
ON KEY LABEL 命令
ON PAD 命令
ON PAGE 命令
ON READERROR 命令

NavigateTo 方法

在一个 Active 文档容器中漫游到指定位置。

语法

Object.NavigateTo(cTarget [, cLocation [, cFrame]])

参数描述

cTarget

指定要定位的 URL。cTarget 可以是一个 URL，或 Active 文档容器（例如 Microsoft Internet Explorer）支持的文档。如果 Active 文档容器不支持 cTarget，则不执行 NavigateTo 方法。

cLocation

指定在 URL 中，或 cTarget 指定的文档中要定位的位置。

cFrame

指定在 URL 中，或 cTarget 指定的文档中要定位的页框。

说明

如果 NavigateTo 方法在一个包含在容器中的 Visual FoxPro Active Document 中执行，则 Active Document 宿主程序定位到指定位置。

如果 NavigateTo 方法在一个包含在容器中的 Visual FoxPro Active Document 之外执行，则启动注册为支持 Active Documents 的应用程序（例如 Microsoft Internet Explorer）。例如，如果您从命令窗口中或 Visual FoxPro 程序 (.prg) 中执行 NavigateTo

方法，则会启动注册的 Active Document 容器，并且定位到指定位置。

附注每当 NavigateTo 方法在一个包含在容器中的 Visual FoxPro Active Document 之外执行时，都会启动注册的 Active Document 容器的一个实例。为了在一个 Visual FoxPro 应用程序中定位到一个 URL，而不启动注册的 Active Document 容器的一个实例，可以使用 Hyperlink Button 基类。

应用于

超链接对象

请参阅

GoBack 方法, GoFoward 方法

NDX() 函数

返回为当前表或指定表打开的某一索引 (.IDX) 文件的名称。

语法

NDX(nIndexNumber [, nWorkArea | cTableAlias])

返值类型

字符型

参数描述

nIndexNumber

指定返回哪一个 .IDX 文件名。USE 和 SET INDEX 命令都支持指定索引文件列表，从而为一个表打开多个 .IDX 文件。索引文件列表中文件名的次序决定了 NDX( ) 函数返回哪一个 .IDX 文件。例如，若 *nIndexNumber* 为 1，则 NDX( ) 返回索引文件列表中的第一个 .IDX 文件；若 *nIndexNumber* 为 2，则 NDX( ) 返回第二个 .IDX 文件，依此类推。NDX( ) 忽略索引文件列表中的复合索引文件 (.CDX)。

如果 *nIndexNumber* 的值大于索引文件列表中的文件数目，NDX( ) 将返回一个空字符串。

nWorkArea

指定的工作区号，返回此工作区的 .IDX 文件。如果指定工作区中没有打开的表，NDX( ) 将返回空字符串。如果省略 *nWorkArea* 参数，NDX( ) 将返回与当前工作区中的表一起打开的 .IDX 文件名。

cTableAlias

指定的表别名，返回此别名的 .IDX 文件。如果没有此表别名，Visual FoxPro 将产生错误信息。如果忽略 *cTableAlias*，NDX( ) 将返回与当前工作区中的表一起打开的 .IDX 文件名。

说明

CDX( )和 MDX( )函数可以返回已打开的复合索引(.CDX)文件名。
在 Visual FoxPro 中，如果 SET FULLPATH 为 ON， NDX( )将返回 .IDX 文件名以及 IDX 文件的路径。若 SET FULLPATH 为 OFF， NDX( )将返回 .IDX 文件名以及 .IDX 文件所在的驱动器。

请参阅

CDX( ), INDEX, MDX( ), SET FULLPATH, SET INDEX, SYS(14), TAG( ), USE

NewIndex 属性

为最新添加在组合框或列表框控件中的项指定索引。设计时不可用，运行时只读。

语法

Control.NewIndex

说明

把项添加到一个已排序列表中时，NewIndex 属性特别有用。

应用于

组合框，列表框

请参阅

Caption 属性，NewItemID 属性，Name 属性

NewItemID 属性

为最新添加在组合框或列表框控件中的项指定标识号。设计时不可用，运行时只读。

语法

[Form.]Control.NewItemID

说明

有关项标识号与项索引之间差异的详细内容，请参阅 AddItem 方法或 AddListItem 方法。

应用于

组合框，列表框

请参阅

AddItem 方法，AddListItem 方法，Clear 方法，ItemData 属性，List 属性，ListCount 属性，MultiSelect 属性，NewIndex 属性，RemoveItem 方法，Selected

NEWOBJECT() 函数

直接从一个 .vcx 可视类库或程序创建一个新类或对象。

语法

NEWOBJECT(*cClassName* [, *cModule* [, *cInApplication* [, *eParameter1* , *eParameter2* , ...]]])

返值类型

对象

参数描述

cClassName

指定类或对象，从该类或对象创建新的类或对象。

对于 cClassName，可使用以下语法创建 OLE 对象：

ApplicationName.Class

例如，为了创建一个 Microsoft Excel 工作表（支持 OLE 自动服务），可以

使用以下语法：

```
oExcelSheet = NEWOBJECT('Excel.Sheet')
```

cModule

指定 .vcx 可视类库或 Visual FoxPro 程序（.prg、.mpr、.app、.exe 等等），其中包含 cClassName 指定的类或对象。默认指定一个 .vcx 可视类库；如果想指定一个程序，必须包含一个扩展名。

注意 类库可以有一个别名。为了使用类库别名指定类库中的类或对象，可在类库后面带一个点号以及对象名。

如果省略 *cModule*，或者 *cModule* 为空字符串或 null 值，则 Visual FoxPro 按以下顺序搜索类或对象：

1. Visual FoxPro 基类。
2. 按加载的顺序在内存中搜索用户自定义类。
3. 当前程序中的类。
4. 使用 SET CLASSLIB 打开的类库。
5. 过程文件中使用 SET PROCEDURE 打开的类。
6. Visual FoxPro 程序执行链中的类。
7. 如果 SET OLEOBJECT 为 ON，则搜索 OLE 注册表。

cInApplication

指定 Visual FoxPro 应用程序（.exe 或 .app），其中包含 *cClassLibName* 指定的 .vcx 可视类库。该应用程序必须具有扩展名。如果省略 *cModule*，或者

cModule 为空字符串或 null 值，则忽略 *cInApplication*。

eParameter1, *eParameter2*, ...

指定可选参数，该参数传递到类或对象的 Init 事件。

说明

NEWOBJECT() 允许您创建一个新类或对象，而不必打开一个 .vcx 可视类库或过程文件。

可以使用 = 或 STORE，将 NEWOBJECT() 返回的对象引用分配给一个变量或数组元素。如果分配给变量或数组元素的一个对象被释放了，则该变量或数组元素包含 null 值。可以使用 RELEASE 命令从内存中删除变量或数组元素。

请参阅

类设计器, CREATE CLASS, CREATE CLASSLIB, CREATEOBJECT( ) , DEFINE CLASS, NewObject **方法**

NewObject 方法

直接从一个 .vcx 可视类库或程序将一个新类或对象添加到一个对象中。

语法

Object.NEWOBJECT(cObjectName, *cClassName* [, *cModule* [, *cInApplication*

[, *eParameter1* , *eParameter2* , ...]]])

cObjectName

指定用于引用新添加类或对象的名称。

cClassName

指定类或对象，从该类或对象添加新的类或对象。

cModule

指定一个 .vcx 可视类库或 Visual FoxPro 程序（.prg、.mpr、.app、.exe 等等），其中包含 cClassName 指定的类或对象。默认的是一个 .vcx 可视类库；如果想指定一个程序，必须包含一个扩展名。

注意 一个类库可以具有别名。为了通过一个别名指定类库中的类或对象，在该类库别名后面加一个点号以及对象的名称。

如果省略 *cModule*，或者 *cModule* 为空字符串或 null 值，则 Visual FoxPro 按以下顺序搜索类或对象：

1. Visual FoxPro 基类。
2. 按加载的顺序在内存中搜索用户自定义类。
3. 当前程序中的类。
4. 使用 SET CLASSLIB 打开的类库。
5. 过程文件中使用 SET PROCEDURE 打开的类。
6. Visual FoxPro 程序执行链中的类。
7. 如果 SET OLEOBJECT 为 ON，则搜索 OLE 注册表。

cInApplication

指定 Visual FoxPro 应用程序（.exe 或 .app），其中包含 *cClassLibName* 指定的 .vcx 可视类库。该应用程序必须具有扩展名。如果省略 *cModule*，或者 *cModule* 为空字符串或 null 值，则忽略 *CinApplication*。

eParameter1, *eParameter2*, ...

指定可选参数，该参数传递到类或对象的 Init 事件。

说明

NEWOBJECT() 允许您创建一个新类或对象，而不必打开一个 .vcx 可视类库或过程文件。

应用于

列，命令按钮组，容器对象，自定义对象，数据环境，表单，表单集，表格，选项按钮组，页面，页框，屏幕，工具栏

请参阅

AddProperty 方法, 类设计器, CREATE CLASS, CREATE CLASSLIB, CREATEOBJECT( ) , DEFINE CLASS, NEWOBJECT( )

NoDataOnLoad 属性

激活一个与临时表相关联的视图，但不下载数据。设计时可用，运行时可读写。

语法

DataEnvironment.Cursor.NoDataOnLoad[= lExpr]

参数描述

lExpr

NoDataOnLoad 属性设置为：

设置	说明
“真”(.T.)	打开与临时表相关联的视图，但不下载数据。
“假”(.F.)	(默认值)打开与临时表相关联的视图，并下载数据。

说明

注意 如果使用 CURSORSETPROP( ) 访问 Cursor 对象，则在运行时

NoDataOnLoad 属性为只读。

设计时可用，运行时可读写。使用 NoDataOnLoad 属性可以确保快速加载与视图相关联的表单集或表单（从后端服务器下载数据可能耗费很长时间）。把 REQUERY() 函数和 NoDataOnLoad 属性一起使用，可以在加载表单之后立即向活动视图下载数据。NoDataOnLoad 属性与 USE 命令 NODATA 子句的行为相似。

应用于

临时表

请参阅

REQUERY(), USE

NORMALIZE() 函数

把用户提供的字符表达式转换为可以与 Visual FoxPro 函数返回值相比较的格式。

语法

NORMALIZE(cExpression)

返值类型

字符型

参数描述

cExpression
指定要规范化的字符表达式。

说明

NORMALIZE( ) 由 *cExpression* 字符表达式返回一个字符串，并做如下变化：

- 字符表达式被转换为大写形式，但嵌入串例外。嵌入串的一个例子是“LEFT('Hello',l)”字符表达式中的 "Hello"。
- 字符表达式中任何缩写的 Visual FoxPro 关键字都被扩展为非缩写形式。
- 把所有分隔字段名与别名的 -> 操作符转换为句点。
- 检查字符表达式中所有的 Visual FoxPro 命令或函数的语法，但并不计算它们的值。若语法不正确，Visual FoxPro 将产生语法错误信息。NORMALIZE( ) 不检查字符表达式中字段、表、变量、用户自定义函数或其他引用是否存在。

例如，用户可在表达式生成器中输入如下索引表达式：

```
UPPE(cust->lname) + UPPE(cust->fname)
```

虽然这是一个有效的 Visual FoxPro 索引关键字表达式，但很难使其与 KEY( ) 这样的 Visual FoxPro 函数的返回值作比较。对于上述表达式，NORMALIZE( ) 将返回下面的字符串：

```
UPPER(CUST.LNAME) + UPPER(CUST.FNAME)
```

这可以很容易地与 KEY( ) 这样的函数返回值相比较。本例子可以判断由具有用户提供索引表达式的索引或索引标识是否存在。

请参阅

EVALUATE( ), ISUPPER( ), PROPER( ), UPPER( )

Note 命令

在一个程序文件中，表明一个非可执行注释行的开始。

语法

NOTE [Comments]

参数描述

Comments

指定注释。

说明

在每个接前行的注释行后面放一个分号 (;)。

示例

```
Note  Initialize the page number;
    variable.
```

```
STORE 1 to gnPageNum
* Set up the loop
DO WHILE gnPageNum <= 25  && 循环 25 次
    gnPageNum = gnPageNum + 1
ENDDO  && 当 gnPageNum <= 25 时，执行操作
```

请参阅

&&, *, MODIFY COMMAND, MODIFY FILE

NTOM() 函数

由数值表达式返回一个含有四位小数的货币值。

语法

NTOM(nExpression)

返值类型

货币型

参数描述

nExpression

指定的一个数值表达式，NTOM()返回其货币值。如果 *nExpression* 的小数位数大于四位，则将其圆整至四位；如果少于四位，则添加 0 补齐四位。

示例

下面的示例创建了一个数值型变量 gnNumeric。TYPE() 显示 N，表示此变量是数值型。用 NTOM() 转换变量为货币型，现在 TYPE() 显示 Y，表示此变量转换后是货币型。

```
STORE 24.95 TO gnNumeric && 创建一个数值型内存变量
CLEAR
? "gnNumeric is type: "
?? TYPE('gnNumeric') && 显示 N, 数值型内存变量
gnNumeric= NTOM(gnNumeric)  && 将数值型值转化为货币型
? "gnNumeric is now type: "
?? TYPE('gnNumeric') && 显示 Y, 货币型值
```

请参阅

MTON()

NullDisplay 属性

指定要显示的文本，该文本表明是 null 值。设计和运行时可用。

语法

Object.NullDisplay[= cNullText]

参数描述

cNullText

指定要显示的文本，该文本表明是 null 值。.NULL. 是默认值。

说明

该属性只对单个的对象更改 null 显示。为了对所有对象更改 null 显示，可使用 SET NULLDISPLAY 命令。

应用于

组合框，编辑框，列表框，微调，文本框

请参阅

CREATE TABLE-SQL, SET NULLDISPLAY

NumberOfElements 属性

指定使用数组中的多少项来填充组合框或列表框控件中的列表部分。设计和运行时可用。

语法

[Form.]Control.NumberOfElements[= nTotal]

参数描述

nTotal

指定一个列表能包含的元素数目。

说明

并且仅当 ListSourceType 属性设置为 5 (数组)，而 ColumnCount 属性设置为 1 时才可用。

要限制列表中显示的数组元素数目时，NumberOfElements 非常有用。列表使用的元素从 FirstElement 属性的设置值开始，包含元素数目是 NumberOfElements 属性的设置值。

应用于

组合框，列表框

请参阅

FirstElement **属性**

NUMLOCK() **函数**

返回 NUM LOCK **键的当前状态，或者设置** NUM LOCK **键的状态为开或关。**

语法

NUMLOCK([lExpression])

返值类型

逻辑值

参数描述

lExpression

打开或关闭 NUM LOCK **键。若** lExpression **为“真”** (.T.)**，则** NUM LOCK **键打开；否则** NUM LOCK **键关闭。在发出** NUMLOCK(.T.) **或** NUMLOCK(.F.) **之前，** NUMLOCK( ) **返回一个代表** NUM LOCK **键设置的逻辑值。**

说明

若 NUM LOCK 键为打开状态，NUMLOCK( )返回“真”(.T.)(在数字键盘上击键将返回一个数字)，否则 NUM LOCK( )返回“假”(.F.)(在数字键盘上击键将移动光标)。

示例

在下例中，用等号(=)来执行 NUMLOCK( )函数，而不返回任何值。

```
gcOldLock = NUMLOCK( ) && 保存初始设置。
WAIT WINDOW ''Press a key to turn Num Lock off''
= NUMLOCK(.T.) && 打开 Num Lock。
WAIT WINDOW 'Press a key to turn Num Lock off'
= NUMLOCK(!NUMLOCK( )) && 设置 NUM LOCK 为原先值的相反值。
WAIT WINDOW 'Press a key to restore original Num Lock Setting'
= NUMLOCK(gcOldLock)  && 返回初始设置。
```

请参阅

CAPSLOCK( ), INSMODE( )

NVL( ) 函数

从两个表达式中返回一个非 null 值。

语法

NVL(eExpression1, eExpression2)

返值类型

字符型、日期型、日期时间型、数值型、货币型、逻辑型或 null 值

参数描述

eExpression1, eExpression2

如果 *eExpression1* 的计算结果为 null 值，则 NVL( ) 返回 *eExpression2* 。如果 *eExpression1* 的计算结果不是 null 值，则返回 *eExpression1* 。*eExpression1* 和 *eExpression2* 可以是任意一种数据类型。如果 *eExpression1* 与 *eExpression2* 的结果皆为 null 值，则 NVL( ) 返回 .NULL. 。

说明

在不支持 null 值或 null 值无关紧要的情况下，可以使用 NVL( ) 来移去计算或操作中的 null 值。

示例

下面的示例创建了一个名为 glMyNull 的变量，它含有 null 值。NVL( ) 用来从 glMyNull 和另一个表达式返回一个非 null 值。

```
STORE .NULL. TO glMyNull && 包含 null 值的变量
CLEAR
? NVL(.T., glMyNull) && 显示 .T.
? NVL(glMyNull, glMyNull) && 显示 .NULL.
```

请参阅

ISNULL(), SET NULL

Object 属性

提供访问 OLE 对象属性和方法的能力，OLE 对象属性和方法由 OLE 服务器提供。设计时不可用。运行时，根据 OLE 服务器来决定 OLE 对象的此属性为只读还是读写。

语法

OLEObject.Object[.Property] [= eValue]

– 或 –

OLEObject.Object[.Method]

参数描述

Property

指定 OLE 服务器提供给 OLE 对象的属性。

eValue

指定 OLE 服务器提供给 OLE 对象的属性值。

Method

指定 OLE 服务器提供给 OLE 对象的方法。

说明

有关 OLE 服务器所支持的属性和方法的详细内容，请参阅有关创建此 OLE 对象的应用程序的文档。例如，如果 OLE 对象是一个 Microsoft Excel 工作簿，那么请参阅有关

Excel 文档，获得 Excel（支持 OLE 的应用程序）所支持的属性与方法。

应用于

OLE 绑定型控件，OLE 容器控件

请参阅

CREATEOBJECT(), GETOBJECT()

对象集合（Object Collection）

一个可以访问 Application 对象中的对象的数组。

语法

ApplicationObject.Objects(nIndex)

参数描述

nIndex

唯一标识 Application 对象中的一个对象。注意 nIndex 可能不对应着对象的创建顺序。

说明

该对象集合可以用来确定一个 Application 对象中的当前对象。对于某些 Visual FoxPro 对象，例如表单，该对象集合可能出现在“调试”窗口中。

应用于

Application 对象，_VFP 系统变量

请参阅

Forms 属性

OBJNUM() 函数

包含此项是为了提供向后兼容性。请使用控件的 TabIndex 属性。

OBJTOCLIENT() 函数

返回一个控制或对象相对于表单的位置或尺寸。

语法

OBJTOCLIENT(ObjectName, nPosition)

返值类型

数值型

参数描述

ObjectName

指定控制或对象名称，返回其表单位置。

nPosition

指定返回对象或控制的哪一个表单位置或尺寸。下表列出了 *nPosition* 的值，以及返回的相应位置或尺寸。

nPosition	位置或尺寸
1	顶边
2	左边
3	宽度
4	高度

说明

OBJTOCLIENT( ) 返回一个控制或对象相对于所在表单客户区的位置或尺寸。例如，一个控制或对象可以放在一个页框中的页面上，而这个页框又放在一个表单上。Top、Left、Width 和 Height 属性将返回控制或对象相对于所在页面的位置和尺寸。但是也可以使用 OBJTOCLIENT( ) 来确定控制或对象相对于页所在表单的位置和尺寸。

OBJTOCLIENT( ) 返回值以像素为单位。

示例

下面的示例用 OBJTOCLIENT( ) 来显示与放置的表单相关联的两个复选框的位置和尺寸。也可用 Top、Left、Width 和 Height 属性来显示与放置的页框相关联的两个复选框的位置和尺寸。

在表单上放置一个命令按钮和页框。PageCount 用来指定页框中页面的数量。将 Tabs 属性设置为“真”(.T.)以指定每页都有选项卡。在每个页面上的不同地点放置复选框。

单击一个复选框时，执行此复选框的 Click 过程。如果选定了此复选框，则 Top、Left、Width 和 Height 属性显示与放置的页框相关联的复选框的位置和尺寸，并且 OBJTOCLIENT()显示与放置的表单相关联的复选框的位置和尺寸。如果没有选定此复选框，会清除 Visual FoxPro 主窗口。

```
CLEAR
STORE _DBLCLICK TO gnDblClick && 保存复选框值
STORE 0.05 TO _DBLCLICK && 使复选框不相似

frmMyForm = CREATEOBJECT('Form') && 创建一个表单
frmMyForm.Closable = .f. && 使窗口 pop-up 菜单不可用

frmMyForm.Move(150,10) && 移动表单

frmMyForm.AddObject('cmbCommand1','cmdMyCmdBtn') && 增加命令按扭
frmMyForm.AddObject('pgfPageFrame1','pgfMyPageFrame') && 增加页边框
frmMyForm.pgfPageFrame1.Page1.AddObject('chkCheckBox1','chkMyCheckBox1')
frmMyForm.pgfPageFrame1.Page2.AddObject('chkCheckBox2','chkMyCheckBox2')
```

```
                                        主窗口
        STORE gnDblClick TO _DBLCLICK  && 恢复复选框
ENDDEFINE

DEFINE CLASS pgfMyPageFrame AS PageFrame  && 创建页边框
    Left = 10  && 页边框列
    Top = 10  && 页边框行
    Height = 175  && 页边框高
    Width = 350  && 页边框高
    PageCount = 2  && 2个页边框
    Tabs = .T.  && 使 Tabs 可见
ENDDEFINE
```

```
DEFINE CLASS chkMyCheckBox1 AS CheckBox  && 创建第一个选项框
    Top = 0
    Width = 200
    Caption = 'Display Position'

PROCEDURE Click
    DO CASE
        CASE ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Value = 0
            ACTIVATE SCREEN
            CLEAR
        CASE ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Value = 1
            ACTIVATE SCREEN
            CLEAR
            ? 'Positions relative'
            ? 'to PageFrame:'
            ?
            ? 'Top: '
            ?? ALLTRIM(STR;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Top))
            ? 'Left: '
            ?? ALLTRIM(STR;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Left))
            ? 'Width: '
            ?? ALLTRIM(STR;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Width))
            ? 'Height: '
            ?? ALLTRIM(STR;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1.Height))
            ?
            ? 'Positions relative'
```

```
            ? 'to Form:'
            ?
            ? 'Top: '
            ?? ALLTRIM(STR(OBJTOCLIENT;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1,1)))
            ? 'Left: '
            ?? ALLTRIM(STR(OBJTOCLIENT;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1,2)))
            ? 'Width: '
            ?? ALLTRIM(STR(OBJTOCLIENT;
                (ThisForm.pgfPageFrame1.Page1.chkCheckBox1,3)))
            ? 'Height: '
            ?? ALLTRIM(STR(OBJTOCLIENT(ThisForm.pgfPageFrame1.Page1.chkCheckBox1,4)))
    ENDCASE
ENDDEFINE

DEFINE CLASS chkMyCheckBox2 AS CheckBox  && 创建第二个选项框
    Top = 30
    Left = 175
    Width = 200
    Caption = 'Display Position'

PROCEDURE CLICK
    DO CASE
        CASE ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Value = 0
            ACTIVATE SCREEN
            CLEAR
        CASE ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Value = 1
            ACTIVATE SCREEN
            CLEAR
```

```
? 'Positions relative'
? 'to PageFrame:'
?
? 'Top: '
?? ALLTRIM(STR(ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Top))
? 'Left: '
?? ALLTRIM(STR;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Left))
? 'Width: '
?? ALLTRIM(STR;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Width))
? 'Height: '
?? ALLTRIM(STR;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2.Height))

?
? 'Positions relative'
? 'to Form:'
?
? 'Top: '
?? ALLTRIM(STR(OBJTOCLIENT;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2,1)))
? 'Left: '
?? ALLTRIM(STR(OBJTOCLIENT;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2,2)))
? 'Width: '
?? ALLTRIM(STR(OBJTOCLIENT;
     (ThisForm.pgfPageFrame1.Page2.chkCheckBox2,3)))
? 'Height: '
?? ALLTRIM(STR(OBJTOCLIENT;
```

```
            (ThisForm.pgfPageFrame1.Page2.chkCheckBox2,4)))
   ENDCASE
ENDDEFINE
```

请参阅

Height 属性, Left 属性, Top 属性, Width 属性

OBJVAR() 函数

包含此项是为了提供向后兼容性。请使用控件的 Name 属性。

OCCURS() 函数

返回字符表达式在另一个字符表达式中出现的次数。

语法

OCCURS(cSearchExpression, cExpressionSearched)

返值类型

数值型

参数描述

cSearchExpression

指定的字符表达式，OCCURS( )在 *cExpressionSearched* 中查找该表达式。

cExpressionSearched

指定的另一个字符表达式，OCCURS( )在其中查找 *cSearchExpression* 的字符表达式。

说明

如果在 *cExpressionSearched* 中没有找到 *cSearchExpression*，OCCURS( )返回 0（零）。

示例

```
STORE 'abracadabra' TO gcstring
CLEAR
? OCCURS('a', gcstring) && 显示数值 5
? OCCURS('b', gcstring) && 显示数值 2
? OCCURS('c', gcstring) && 显示数值 1
? OCCURS('e', gcstring) && 显示数值 0
```

请参阅

$, INLIST( )

OEMTOANSI() 函数

包含此项是为了提供向后兼容性。请使用 GETCP() 函数。

OLDVAL() 函数

返回字段的初始值，该字段值已被修改但还未更新。

语法

OLDVAL(cExpression [, cTableAlias | nWorkArea])

返值类型

字符型、货币型、日期型、日期时间型、双精度型、浮点型、整型、逻辑型、数值型或备注型。

参数描述

cExpression

指定一表达式，OLDVAL()函数从表或远程数据源中返回其初始值。一般情

况下，*cExpression* 是一个字段，或者是一个表达式，此表达式包含了表或远程数据源中的一系列字段。

cTableAlias
指定表或临时表的别名，从中返回字段初始值。

nWorkArea
指定表或临时表所在的工作区，从中返回字段初始值。

说明

OLDVAL( ) 函数返回 Visual FoxPro 表或临时表中记录字段的初始值，必须用 CURSORSETPROP( ) 命令对该表启用行缓冲或表缓冲。
如果数据库中的表或临时表具有有效性规则，那么 OLDVAL( ) 返回字段初始值时不必启用行缓冲或表缓冲。
如果在启用行缓冲的情况下把记录指针移到另外的记录上，或发出 TABLEUPDATE( ) 命令执行对记录的更改，或者发生了结束一个事务这样的行为而引起更新时，将更新字段，其初始值不再可用。
OLDVAL( ) 返回的数据类型取决于 *cExpression* 所指定的表达式。
如果不带 *cTableAlias* 或 *nWorkArea* 参数而发出 OLDVAL( )，则返回当前选定工作区中所打开表或临时表字段的初始值。

示例

下面的示例演示了如何用 OLDVAL( ) 返回一个缓冲表中的字段的最初值。创建表 employees 并用 INSERT – SQL 在 cLastName 字段中插入值 “Smith”。

MULTILOCKS 设置为 ON，这是表缓冲所要求的。用 CURSORSETPROP() 将缓冲模式设置为最好的表缓冲 (5)。
显示 cLastName 字段 (Smith) 的最初值，然后用 REPLACE 修改 cLastName 字段。显示 cLastName 字段 (Jones) 的新值。用 OLDVAL( ) 显示 cLastName 字段 (Smith)的最初值。然后用 TABLEUPDATE( ) 提交更改到表中。接着显示 cLastName 字段 (Jones) 更新后的值。

```
CLOSE DATABASES
CLEAR

*创建新表并增加一条空记录
CREATE TABLE employee (cLastName C(10))
APPEND BLANK

*插入初始值
INSERT INTO employee (cLastName) VALUES ("Smith")

*设置表的缓冲区
SET MULTILOCKS ON  &&允许表缓冲
=CURSORSETPROP("Buffering", 5, "employee" )  &&设置表的缓冲

*显示初始值
=MESSAGEBOX("Original cLastName value: "+ cLastName, 0, "Results")
* Change record value and display results
REPLACE cLastName WITH "Jones"
=MESSAGEBOX("Modified cLastName value: "+ cLastName, 0, "Results")

*保存字段初值到 cTemp 变量并显示结果
```

```
cTemp=OLDVAL("cLastName", "employee")
=MESSAGEBOX("Original cLastName value: "+ cTemp, 0, "Results")

*更新表并显示终值
=TABLEUPDATE(.T.)
=MESSAGEBOX("Final cLastName value: "+ cLastName, 0, "Results")

* 关闭并删除样表文件
USE
DELETE FILE employee.dbf
```

请参阅

CURVAL( ), GETFLDSTATE( ), TABLEREVERT( ), TABLEUPDATE( ), CURSORSETPROP( )

OLE Bound 控件

创建一个 OLE 绑定型控件。

语法

OLEBoundControl

说明

在表单或报表中，一个 OLE 绑定型控件允许在表中的通用字段上显示一个 OLE 对象（例如来源于 Microsoft Word 或 Microsoft Excel）的内容。

与 OLE Container 控件不同，可插入的 OLE 对象没有自己的事件集合。同样，OLE 绑定型控件也与 OLE Container 控件不同，它与一个 Visual FoxPro 表的通用字段连接在一起。

有关 Visual FoxPro 中 OLE 对象的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十六章“添加 OLE”。

属性

Application	AutoActivate	AutoSize
BaseClass	ClassLibrary	Class
Comment	ControlSource	DocumentFile
DragIcon	DragMode	Enabled
Height	HelpContextId	HostName
Left	MousePointer	Name
Object	OLEClass	OLELCID
OLETypeAllowed	ParentClass	Sizable
Stretch	TabIndex	TabStop

Tag　　Top　　Value
Visible　　WhatsThisHelpID　　Width

事 件

Destroy　　DragDrop　　DragOver
Error　　GotFocus　　Init
LostFocus　　Moved　　Resize
UIEnable

方 法

AddProperty　　CloneObject　　DoVerb
Drag　　Move　　Refresh
ResetToDefault　　SaveAsClass　　SetFocus
Zorder

请 参 阅

APPEND GENERAL, CREATE FORM, CREATE CLASS, CREATEBINARY( ), DEFINE CLASS, MODIFY GENERAL, OLE

OLE Container 控件

 创建一个 OLE 容器控件。

语法

OLEControl

说明

OLE 容器控件允许向应用程序中加入 OLE 对象。OLE 对象包括 ActiveX 控件（.OCX 文件），或者其他应用程序（例如 Microsoft Word 和 Microsoft Excel）创建的可插入 OLE 对象。与 ActiveX 控件（.OCX 文件）不同的是，可插入 OLE 对象没有自己的事件集合。OLE 容器控件与 OLE 绑定型控件也不同，它不与 Visual FoxPro 表的一个通用字段相连接。

附注放在 OLE 容器控件上的 ActiveX 控件的类型决定了这个 ActiveX 控件可用的属性、事件和方法。

有关 Visual FoxPro 中 OLE 对象的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十六章“添加 OLE”。

属性

Align
Application
AutoActivate
AutoSize
AutoVerbMenu
BaseClass
Cancel
Class
ClassLibrary
Comment
ControlSource
Default
DocumentFile
DragIcon
DragMode
Enabled
Height
HelpContextID
HostName
Left
MouseIcon
MousePointer
Name
Object
OLEClass
OLELCID
OLETypeAllowed
Parent
ParentClass
Sizable
Stretch
TabIndex
TabStop
Tag
Top
Visible
WhatsThisHelpID
Width

事件

Destroy
DragDrop
DragOver
Error
GotFocus
Init
LostFocus
Moved
Resize
UIEnable

方法

AddProperty	CloneObject	DoVerb
Drag	Move	Refresh
ResetToDefault	SaveAsClass	SetFocus
ShowWhatsThis	Zorder	

示例

下面的示例在表单中添加一个 OLE 容器控件，并且用 OleClass 和 DocumentFile 属性来指定 Microsoft Excel 作为 Automation 服务程序和一个 Excel 工作表作为编辑的文件。

DocumentFile 属性指定了驱动器 C 下的 EXCEL 目录中的工作表文件 Book1.xls。如果 DocumentFile 属性中指定的文件和目录不存在，本示例将不能正常运行，可能需要修改 DocumentFile 属性以指定一个现有的目录和工作表文件。

* DoVerb 方法被用来激活待编辑的工作表。

```
frmMyForm = CREATEOBJECT('Form') && 创建一个表单
frmMyForm.Closable = .F. && 使窗口的 pop-up 菜单不可用

frmMyForm.AddObject('cmdCommand1','cmdMyCmdBtn') && 增加命令按扭
frmMyForm.AddObject("oleObject","oleExcelObject") && 增加 OLE 对象

frmMyForm.cmdCommand1.Visible=.T. && 显示 "退出" 命令按扭

frmMyForm.oleObject.Visible=.T. && 显示 OLE 控件
frmMyForm.oleObject.Height = 50 && OLE 控件高
```

```
frmMyForm.Show  && 显示表单

frmMyForm.oleObject.DoVerb(-1)  && -1 for Edit

READ EVENT  && 启动事件程序

DEFINE CLASS oleExcelObject as OLEControl
    OleClass ="Excel.Sheet"  && 服务器名称
    DocumentFile = "C:\EXCEL\BOOK1.XLS"  && 存在此文件
ENDDEFINE

DEFINE CLASS cmdMyCmdBtn AS CommandButton  && 创建命令按扭
    Caption = '\<Quit'  && 给命令按扭增加说明
    Cancel = .T.  && 默认取消命令按扭(Esc键)
    Left = 125  && 命令按扭列
    Top = 210  && 命令按扭行
    Height = 25  && 命令按扭高

    PROCEDURE Click
       CLEAR EVENT  && 终止事件程序，关闭表单
ENDDEFINE
```

请参阅

ActiveX 控件概览, APPEND GENERAL 命令, CREATE FORM 命令, CREATE CLASS 命令, CREATEBINARY(), DEFINE CLASS 命令, MODIFY GENERAL 命令, OLE 绑定式控件

OLEClass 属性

返回一个 OLE 对象的命名类 ID。对于一个现有对象。在设计时和运行时只读，但是可以在创建一个对象时设置该对象的 OLEClass 属性。

语法

Control.OLEClass[= cName]

参数描述

cName

对象的命名类 ID。这是用于创建该对象的应用程序的注册名，或者是激活该对象时激活的应用程序的注册名。

说明

当开始向一个表单添加一个 OLE 容器时，可以使用“插入对象”对话框，设置该 OLE 容器对象的 OLEClass 属性，或者在类定义中创建一个 OLE 对象时通过代码设置该 OLE 对象的 OLEClass 属性。当使用 APPEND GENERAL 命令创建 OLE 对象时，也可以设置这个属性。

一个 OLE 对象的 OLEClass 属性指定了用于创建或编辑该对象的应用程序。若要指定该对象的真正内容，可设置它的 DocumentFile 属性。

示例

下面的示例在表单中添加一个 OLE 容器控件，并且用 OleClass 和 DocumentFile 属性来指定 Microsoft Excel 作为 Automation 服务程序和一个 Excel 工作表作为编辑的文件。

```
Define class foo as form
    add object oleXLSheet1 as oleXLSheet
EndDefine

Define class oleXLSheet as OLECONTROL
    oleclass = "Excel.Sheet"
    documentfile="C:\msoffice\Excel\mysheet.xls"
    oletypeallowed = 1  && 嵌入
EndDefine
```

应用于

OLE 绑定型控件，OLE 容器控件

请参阅

APPEND GENERAL, CREATEOBJECT(), DocumentFile 属性

OLECompleteDrag 事件

当数据放落到目标上，或取消了 OLE 拖放操作时发生。

语法

PROCEDURE Object.OLECompleteDrag
LPARAMETERS nEffect

参数描述

nEffect

从 OLEDragDrop 事件传递的值，对应于当数据放落到目标上时采取的动作。表列出了 nEffect，以及每种动作的说明。

nEffect	Foxpro.h 常数	说明
0	DROPEFFECT_NONE	放落目标不接受数据，或者取消了 OLE 放落操作。
1	DROPEFFECT_COPY	数据从拖动源复制到放落目标中。
2	DROPEFFECT_MOVE	数据从拖动源移动到放落目标中。
4	DROPEFFECT_LINK	数据从拖动源链接到放落目标。

说明

OLECompleteDrag 是一个拖动源事件，并且是 OLE 拖放操作中最后发生的事件。包含 NODEFAULT 可以防止在文本移动过程中删除文本。

这个事件允许拖动源确定对放落到目标上的数据采取的动作。放落目标可以在 OLEDragDrop 事件中设置 nEffect，并且拖动源可以根据 nEffect 的值采取相应的动作。例如，如果数据移动到放落目标上，拖动源应该从自己这里删除这些数据。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览，OLEDragDrop 事件，OLEDropMode 属性

OLEDrag 方法

开始一次 OLE 拖放操作。

语法

PROCEDURE Object.OLEDrag
LPARAMETERS *lDetectDrag*

参数描述

lDetectDrag

指定何时发生 OLEStartDrag **事件。如果** *lDetectDrag* **为“假”** (.T.)**，则立即发生** OLEStartDrag **事件。如果** *lDetectDrag* **为“真”** (.T.)**，则当用户按住鼠标一段时间之后，或者按住鼠标并移动了一段距离之后，发生** OLEStartDrag **事件。**

如果不为 lDetectDrag **指定一个值，则自动返回“真”** (.T.)。

说明

OLEDrag **是一个拖动源方法。**

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop **概览**, OLEDragMode **属性**, OLEStartDrag **事件**

OLEDragDrop 事件

当数据放落到目标上，并且放落目标的 OLEDropMode 属性设置为 1 – Enabled 时发生。

语法

```
PROCEDURE Object.OLEDragDrop
LPARAMETERS oDataObject, nEffect, nButton, nShift,
  nXCoord, nYCoord
```

参数描述

oDataObject

对 OLE 拖放 DataObject 的一个对象引用，同 GetData 和 GetFormat 方法一起使用可以返回 DataObject 中的数据和数据格式。

nEffect

传递给 OLECompleteDrag 事件的一个值，表明当数据放落到目标上时采取的动作。nEffect 的初始值表明拖动源支持的 OLE 拖放操作。在 OLEDragDrop 事件中，您可以更改 nEffect 的值，该值传递给 OLECompleteDrag 事件。下表列出了 nEffect 的值，以及每种动作的说明。

nEffect	Foxpro.h 常数	说明
0	DROPEFFECT_NONE	放落目标不接受数据，或者取消了 OLE 放落操作。
1	DROPEFFECT_COPY	数据从拖动源复制到放落目标中。
2	DROPEFFECT_MOVE	数据从拖动源移动到放落目标中。
4	DROPEFFECT_LINK	数据从拖动源链接到放落目标。

nButton

包含一个数字，该数字指定了为将数据放落到目标上，放开了哪个鼠标键：1（左键）、2（右键）或 4（中间键）。

nShift

包含一个数字，该数字指定了将数据放落到目标上时，辅助键的状态。有效的辅助键是 SHIFT、CTRL 和 ALT。下表列出了对应于单个辅助键的 nShift 返回值。

nShift	辅助键
1	SHIFT
2	CTRL
4	ALT

如果当按下鼠标时按住了多个辅助键，则 nShift 参数包含这些辅助键的值的和。例如，释放鼠标键时按下 CTRL 键，nShift 参数包含 2。但是如果在释放鼠标键时按下 CTRL+ALT， nShift 参数包含 6。

nXCoord, nYCoord

包含当释放鼠标将数据放落到目标上时，鼠标指针在表单中的水平 (nXCoord) 和垂直 (nYCoord) 位置。这些坐标是按表单的坐标系表达的，表单的 ScaleMode 属性指定了度量单位。

说明

OLEDragDrop 是一个放落目标事件，只当控件或对象的 OLEDropMode 属性设置为 1 – Enabled 时发生。如果 OLEDropMode 属性设置为 0 – Disabled 或 2 – Pass to Container，则不发生该事件。

如果在 OleDragDrop 事件中执行自己的放落过程，则包含 NODEFAULT 可以阻止默认放落过程的发生。这时，必须设置 nEffect 的值。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览, OLECompleteDrag 事件, OLEDropMode 属性

OLEDragMode 属性

指定一次拖动操作是如何开始的。设计和运行时可用。

语法

Object.OLEDragMode[= nValue]

参数描述

nValue

指定控件或对象如何处理 OLE 拖动操作。下表列出了 nValue 的设置。

设置	说明
0	人工（默认值）。 当试图拖动数据时，Visual FoxPro 不开始 OLE 拖动操作。要开始 OLE 拖动操作，必须调用 OLEDrag 方法。 如果不包含附加的拖动操作代码，将 OLEDragMode 设置为 0 为现有应用程序提供了向后兼容性（不支持 OLE 拖动）。
1	自动。 当试图拖动数据时，Visual FoxPro 自动开始 OLE 拖动操作。

说明

OLEDragMode 是一个拖动源属性。如果一个控件或对象的 DragMode 属性设置为 1 – 自动，则该控件或对象就不能作为一个 OLE 拖动源。这为以前版本的 Visual FoxPro 对拖放的支持提供了向后兼容性。

注意 表格控件的 OLEDragMode 属性是只读的。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

DragMode 属性，OLE Drag-and-Drop 概览，OLEDrag 方法，OLEStartDrag 事

件

OLEDragOver 事件

当数据拖动到一个放落目标上，并且放落目标的 OLEDropMode 属性设置为 1 – 启用时发生该事件。

语法

```
PROCEDURE Object.OLEDragOver
LPARAMETERS oDataObject, nEffect, nButton, nShift,
  nXCoord, nYCoord, nState
```

参数描述

oDataObject

对 OLE 拖放 DataObject 的一个对象引用，同 GetData 和 GetFormat 方法一起使用可以返回 DataObject 中的数据和数据格式。

nEffect

传递给 OLEGiveFeedback 事件的一个值，表明当数据放落到目标上时采取的动作。nEffect 的初始值表明拖动源支持的 OLE 拖放操作。下表列出了 nEffect 的值，以及每种动作的说明。

nEffect	Foxpro.h constant	说明
0	DROPEFFECT_NONE	放落目标不接受数据，或者取消了 OLE 放落操作。
1	DROPEFFECT_COPY	数据从拖动源复制到放落目标中。
2	DROPEFFECT_MOVE	数据从拖动源移动到放落目标中。
4	DROPEFFECT_LINK	数据从拖动源链接到放落目标。

nButton

包含一个数字，该数字指定了为将数据放落到目标上，放开了哪个鼠标键：1（左键）、2（右键）或 4（中间键）。

nShift

包含一个数字，该数字指定了将数据放落到目标上时，辅助键的状态。有效的辅助键是 SHIFT、CTRL 和 ALT。下表列出了对应于单个辅助键的 nShift 返回值。

nShift	辅助键
1	SHIFT
2	CTRL
4	ALT

如果当按下鼠标时按住了多个辅助键，则 nShift 参数包含这些辅助键的值的和。例如，释放鼠标键时按下 CTRL 键，nShift 参数包含 2。但是如果在释放鼠标键时按下 CTRL+ALT，nShift 参数包含 6。

nXCoord,nYCoord

包含当释放鼠标将数据放落到目标上时，鼠标指针在表单中的水平 (nXCoord) 和垂直 (nYCoord) 位置。这些坐标是按表单的坐标系表达的，表单的 ScaleMode 属性指定了度量单位。

nState

包含一个数值，该数值指定了数据被拖动的方向 – 到控件或对象中、到控件或对象的内部，或到控件或对象的外部。下表列出了 nState 的值。

nState	说明
0	数据被拖动到控件或对象中。当 nState 为零时，可以设置 OLEDropEffects 和 OLEDropHasData 属性。
1	数据被拖动到控件或对象的外部。
2	数据被拖动到控件或对象的内部，

说明

OLEDragOver 是一个放落目标的事件，只有当控件或对象的 OLEDropMode 属性设置为 1 – 启用时才发生。如果 OLEDropMode 属性设置为 0 – 禁止或 2 – 传递到容器，则不发生这个事件。

注意 应该避免在 OLEDragOver 事件中使用 WAIT WINDOW 和 MESSAGEBOX() 这样的命令和函数创造等待状态。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览，OLEGiveFeedBack 事件，OLEDropEffects 属性，OLEDropHasData 属性，OLEDropMode 属性

OLEDragPicture 属性

指定在 OLE 拖放操作过程中，在鼠标指针下显示的图片。设计和运行时可用。

语法

Object.OLEDragPicture[= cFileName]

参数描述

cFileName

指定在 OLE 拖放操作时所显示图片的文件名和路径。可以指定的图形文件有 .bmp、.dib、.jpg、.gif、.ani、.cur 或 .ico。

该图形文件通常是被拖动对象的半透明或轮廓代表。若要创建对象的半透明代表，可使用一个黑白方格掩码文件 (.msk)。

说明

OLEDragPicture 是一个拖动源属性。当鼠标指针放到包含拖动源的表单上时显示该图片。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览

OLEDropEffects 属性

指定一个 OLE 放落目标所支持的放落操作类型。设计和运行时可用。

语法

Object.OLEDropEffects[= nDropEffect]

参数描述

nDropEffect

指定一个 OLE 放落目标所支持的放落操作类型。下表列出了 nDropEffect 的值，以及每个值的说明。

nDrop Effect	Foxpro.h 常数	说明
0	DROPEFFECT_NONE	放落目标不接受数据。
1	DROPEFFECT_COPY	数据可以复制到放落目标中。

续表

2	DROPEFFECT_MOVE	数据可以移动到放落目标中。
4	DROPEFFECT_LINK	数据可以链接到放落目标。

通过为 nDropEffect 添加多个值，一次放落可以支持多种放落操作。例如，如果 nDropEffect 为 3，则放落目标支持复制和移动放落操作。

说明

OLEDropEffects 是一个放落目标属性，应该在 OLEDragOver 事件中设置。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览，OLEDragOver 事件，OLEDropHasData 属性

OLEDropHasData 属性

指定如何管理一次放落操作。运行时可用，设计时只读。

语法

Object.OLEDropHasData[= nDropEffect]

参数描述

nDropEffect

指定如何管理一次放落操作。下表列出了 nDropEffect 的值，以及每个值的说明。

NDrop Effect	Foxpro.h 常数	说明
	DROPHASDATA_VFPDETERMINE	默认值。Visual FoxPro 自动确定 DataObject 是否包含可以放落到目标上的数据格式的数据。 如果 DataObject 包含的数据的格式适于放落目标，则该数据就放落到目标上。Visual FoxPro 管理鼠标指针，并且通知拖动源。 如果 DataObject 包含的数据的格式不适于放落目标，则 Visual FoxPro 显示不准放落的鼠标指针，取消这次放落操作，并且通知拖动源这次放落操作被取消了。

续表

DROPHASDATA_NOTUSEFUL	DataObject 不包含可以放落到目标上的数据格式的数据，并且显示不准放落的鼠标指针。
DROPHASDATA_USEFUL	DataObject 包含可以放落到目标上的数据格式的数据。

说明

OLEDropHasData 是一个放落目标属性，并且应该在 OLEDragOver 事件中设置。在 OLEDragOver 事件中使用 GetFormat 属性，可以判断 DataObject 是否包含适合于放落目标的数据格式的数据。如果数据符合放落目标的数据格式，则将 OLEDropHasData 设置为 1。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览，OLEDragOver 事件，OLEDropEffects 属性

OLEDropMode 属性

指定放落目标如何管理 OLE 放落操作。设计和运行时可用。

语法

Object.OLEDropMode[= nValue]

参数描述

nValue

指定控件或对象如何管理 OLE 放落操作。下表列出了 nValue 的设置。

nValue	Foxpro.h 常数	说明
0	DROP_DISABLED	禁止（默认值）。 数据不能放落到控件或对象上，并且不发生放落目标事件。当鼠标指针位于控件或对象上时，显示不准放落鼠标指针。 将 OLEDropMode 设置为 0 为现有应用程序提供了向后兼容性（不支持 OLE 放落）。
1	DROP_ENABLED	启用。 Visual FoxPro 允许将数据放落到控件或对象上，并且发生放落目标事件。
2	DROP_PASSTOCONTAINER	传递到容器。 控件或对象就象禁止了 OLE 放落操作一样，并且数据被放落到控件或对象的容器中。 为了接受数据，容器的 OLEDropMode 属性必须设置为 1 或 2。如果容器的 OLEDropMode 属性设置为 0，则显示不准放落鼠标指针。

说明

OLEDropMode 是一个放落目标属性。如果一个控件或对象的 DropMode 属性设置为 0 – 禁止，则该控件或对象不能作为一个 OLE 放落源。这为以前版本 Visual FoxPro 对拖放的支持提供了向后兼容性。

如果一个控件或对象的 Enabled 属性设置为“假” (.F.)，就不能将数据放落到该控件或对象上，当进行放落操作时，会显示不准放落鼠标指针。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

Enabled 属性，OLE Drag-and-Drop 概览

OLEDropTextInsertion 属性

指定是否可以在一个控件的文本框部分的一个单词中间放落文本。

语法

Object.OLEDropTextInsertion[= lExpression]

参数描述

lExpression

OLEDropTextInsertion **属性的设置为：**

设置	说明
0	（默认值）文本可以放落到单词中。
1	文本只能放落到单词的前面或后面。

说明

为了在一个控件的文本框部分放落文本，该控件的 OLEDropMode **属性必须设置为** 1 – **启用。**

应用于

组合框，编辑框，微调，文本框

请参阅

OLE Drag-and-Drop **概览，** OLEDropMode **属性**

OLEGiveFeedback 事件

在每次 OLEDragOver 事件之后发生。允许拖动源指定 OLE 拖放操作的种类以及可视反馈。

语法

PROCEDURE Object.OLEGiveFeedback
LPARAMETERS *nEffect, eMouseCursor*

参数描述

nEffect

当数据放落到目标上时所进行的操作。*nEffect* 的值是在放落目标的 OLEDragOver 事件中设置的。下表列出了 nEffect 的值，以及对每种操作的说明。

nEffect	Foxpro.h 常数	说明
0	DROPEFFECT_NONE	放落目标不接受数据。
1	DROPEFFECT_COPY	放落之后复制。
2	DROPEFFECT_MOVE	放落之后移动。
4	DROPEFFECT_LINK	放落之后链接。

EMouseCursor

指定在 OLE 拖放操作过程中所显示的鼠标指针。*EMouseCursor* 可以是一个字符或数值。EMouseCursor 是一个输出参数，并且在事件的入口设置为零。

如果 *eMouseCursor* 是一个字符值，则假定该字符值是 .ani、.cur 或 .ico 图形文件的名称。如果 *eMouseCursor* 是一个数值，则该值指定了所显示的鼠标指针。下表列出了 *eMouseCursor* 的数值，以及每种鼠标指针的说明。

Emouse Cursor	Foxpro.h 常数	说明
0	MOUSE_DEFAULT	（默认值）对象所确定的形状。
1	MOUSE_ARROW	箭头。
2	MOUSE_CROSSHAIR	十字。一个十字指针。
3	MOUSE_IBEAM	I 型杆。
4	MOUSE_ICON_POINTER	图标。黑块中有一个小白块。

续表

5	MOUSE_SIZE_POINTER	大小柄。
6	MOUSE_SIZE_NE_SW	指向东西南北的四方向箭头。 东北西南大小柄。
7	MOUSE_SIZE_N_S	指向东北和西南的双箭头。 南北大小柄。
8	MOUSE_SIZE_NW_SE	指向南北的双箭头。 西北东南大小柄。
9	MOUSE_W_E	指向西北和东南的双箭头。 东西大小柄。
10	MOUSE_UP_ARROW	指向东西的双箭头。 向上箭头。
11	MOUSE_HOURGLASS	沙漏。
12	MOUSE_NO_DROP	不准放落。
13	MOUSE_HIDE_POINTER	隐藏指针。
14	MOUSE_ARROW2	箭头。
15	MOUSE_ARROW_HOURGLASS	箭头和沙漏。
16	MOUSE_ARROW_QUESTION	箭头和问号。

说明

OLEGiveFeedback 是一个拖动源事件，允许为用户提供可视反馈。也可以更改鼠标指针，以便当鼠标放在拖动源或放落目标上时表明所发生的操作。包含 NODEFAULT 对这个方法的行为没有影响。

附注 您应该避免在 OLEGiveFeedback 事件中使用 WAIT WINDOW 和 MESSAGEBOX() 这样的命令和函数创造等待状态。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，项目挂接对象，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览，OLEDragOver 事件

OLELCID 属性

返回 OLE 绑定型控件或 OLE 容器控件的 OLE Locale ID 值。设计时和运行时只读。

语法

Control.OLELCID

说明

当 OLE 绑定型控件或 OLE 容器控件放置在表单或 Visual FoxPro 主窗口中时，它们的环境 ID 值由表单或 Visual FoxPro 主窗口的 DefOLELCID 属性决定。

当 OLE 绑定型控件或 OLE 容器控件放置在表单或 Visual FoxPro 主窗口中时，表单或 Visual FoxPro 主窗口的 DefOLELCID 属性为 0。控件使用当前的 Visual FoxPro 环境 ID 值 (LCID)。使用 SYS(3004) 可以决定当前的 Visual FoxPro 环境 ID 值。使用 DefOLELCID 属性可以指定表单的环境 ID 值。

有关环境 ID 值的列表，请参阅 SYS(3005)。

注意 OLELCID 属性仅影响 OLE 控件显示的用户接口语言，而不是 OLE 自动化命令语言。OLE 自动化命令语言仅受全局环境 ID 值的影响，请参阅 SYS(3005)。

应用于

OLE 绑定型控件，OLE 容器控件

请参阅

DefOLELCID 属性，SYS(3005)

OLERequestPendingTimeout 属性

指定在 Automation 请求之后多长时间显示一条忙信息。运行时可读写。

语法

ApplicationObject.OLERequestPendingTimeout[= nMilliseconds]

参数描述

nMilliseconds

指定在 Automation 请求之后，显示一条忙信息之前必须经过的毫秒数。当发生一个鼠标或键盘事件时显示该忙信息。NMilliseconds *的*默认值是 5,000 毫秒。

如果 nMilliseconds 为 0，当一个 Automation 请求悬而未决时，并且发生一个鼠标或键盘事件时，不显示忙信息。

应用于

Application 对象，_VFP 系统变量

请参阅

OLEServerBusyRaiseError 属性，OLEServerBusyTimeout 属性

OLEServerBusyRaiseError 属性

指定当一个 Automation 请求被拒绝时是否显示一条错误信息。运行时可读写。

语法

ApplicationObject.OLEServerBusyRaiseError[= lExpression]

参数描述

lExpression

为以下设置：

设置	说明
.T.（真）	当经过 OLEServerBusyTimeout 属性指定的毫秒之后没有发生一个错误，并且不显示一条忙信息。
.F.（假）	（默认值）当经过 OLEServerBusyTimeout 属性指定的毫秒之后发生了一个错误，并且显示一条忙信息。

应用于

Application 对象，_VFP 系统变量

请参阅

OLERequestPendingTimeout 属性，OLEServerBusyTimeout 属性

OLEServerBusyTimeout 属性

指定当一个服务程序忙时，经过多久重新尝试一个 Automation 请求。运行时可读写。

语法

ApplicationObject.OLEServerBusyTimeout[= nMilliseconds]

参数描述

nMilliseconds

指定当显示一条服务程序忙的信息之前，经过多少毫秒重新尝试一个 Automation 请求。

应用于

Application 对象，_VFP 系统变量

请参阅

OLERequestPendingTimeout 属性，OLEServerBusyRaiseError 属性

OLESetData 事件

当放落目标调用 GetData 方法，并且在没有指定格式的数据时，拖动源发生该事件。

语法

PROCEDURE Object.OLESetData
LPARAMETERS *oDataObject, eFormat*

参数描述

oDataObject

对 OLE 拖放 DataObject 的一个对象引用，同 SetData 方法一起使用可以将数据放在 DataObject 中。

eFormat

一个数值或字符值，表明 GetData 方法所需的数据的格式。拖动源使用这个值确定放在 DataObject 中的数据的格式。有关详细内容，请参阅 GetData 方法。

说明

OLESetData 是一个拖动源事件。包含 NODEFAULT 对这个方法的行为没有影响。

当执行 GetData 方法，并且 DataObject 中没有指定（在 GetData 方法中指定）格式的

数据时，发生 OLESetData 事件。在 OLESetData 事件中，可以使用 SetData 方法按指定格式放置 DataObject 中的数据。这个技术称为“延迟呈递”，允许在要求时将数据放在 DataObject 中。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

GetData 方法，OLE Drag-and-Drop 概览，OLESetData 事件，SetData 方法

OLEStartDrag 事件

当调用 OLEDrag 方法时发生。

语法

PROCEDURE Object.OLEStartDrag
LPARAMETERS *oDataObject, nEffect*

参数描述

oDataObject

对 OLE 拖放 DataObject 的一个对象引用。在 OLEStartDrag 事件中，可以调用任何 DataObject 方法。

nEffect

拖动源支持的 OLE 拖动操作。下表列出了 nEffect 的值，以及每种动作的说

明。nEffect 是一个输出参数，并且在事件的入口设置为 3，所以您应该在本事件中提供 nEffect 的值。例如，如果只允许复制操作，可将 nEffect 设置为 1 (DROPEFFECT_COPY)。

nEffect	Foxpro.h constant	说明
0	DROPEFFECT_NONE	拖动源不支持任何拖动操作。
1	DROPEFFECT_COPY	拖动源支持复制操作。
2	DROPEFFECT_MOVE	拖动源支持移动操作（默认的）。
4	DROPEFFECT_LINK	拖动源支持链接操作。

一个拖动源可以支持多个拖动操作，只需将 nEffect 的多个值加在一起。例如，如果 nEffect 为 3，则拖动源支持复制和移动拖动操作 (3 = 1 (复制) + 2 (移动))。

说明

OLEStartDrag 是一个拖动源事件。包含 NODEFAULT 对这个方法的行为没有影响。

应用于

复选框，组合框，命令按钮，命令按钮组，容器对象，控件对象，编辑框，表单，表格，图像，标签，线条，列表框，选项按钮，选项按钮组，页面，页框，形状，微调，文本框，工具栏

请参阅

OLE Drag-and-Drop 概览, OLEDrag 方法, OLEDragMode 属性

OLETypeAllowed 属性

返回控件中所包含 OLE 对象的类型（嵌入或链接）。设计和运行时只读。

语法

Control.OLETypeAllowed[= nValue]

参数描述

nValue

返回一个值，它标明包含于控件中的 OLE 对象的类型（嵌入或链接）。可能的值为 0（链接）、1（嵌入）、-1（不包含 OLE 对象的 OLE 绑定型控件）和 -2（OLE 控件 (.OCX)）。

说明

仅当为 DocumentFile 属性设置时，OLETypeAllowed 属性才有用。

注意 可以使用 DEFINE CLASS 命令在代码中设置 OLETypeAllowed 属性，从而创建一个链接的 OLE 对象。

应用于

OLE 绑定型控件，OLE 容器控件

请参阅

DEFINE CLASS, DocumentFile 属性

ON BAR 命令

指定从菜单中选择特定菜单项时激活的菜单或菜单栏。

语法

ON BAR nMenuItemNumber OF MenuName1
[ACTIVATE POPUP MenuName2
|ACTIVATE MENU MenuBarName]

参数描述

nMenuItemNumber OF MenuName1

指定菜单项的编号及所在菜单，该菜单项将激活其他菜单或菜单栏。菜单中的每一项都可以拥有指定给它的另外一个菜单或菜单栏。

凡拥有指定菜单或菜单栏的菜单项右部都有一个箭头。该箭头表示选中此项时将激活额外的菜单或菜单栏。如果使用 DEFINE POPUP ... MARGIN 定义菜单，则在每一菜单项上为层叠式子菜单箭头留出空间。若在创建菜单时不使用 MARGIN 子句，层叠式子菜单箭头可能会改写菜单项最后一个字符。

ACTIVATE POPUP MenuName2

指定选择该菜单项时激活的菜单名称。使用不带 ACTIVATE POPUP 的 ON BAR 命令，可以从菜单项中释放菜单。

ACTIVATE MENU MenuBarName

指定选择该菜单项时激活的菜单栏名称。使用不带 ACTIVATE POPUP 的 ON BAR 命令，可以从菜单项中释放菜单栏。

说明

一个可以显示和激活其他菜单的菜单称为层叠式子菜单。当选中菜单中的菜单项时，使用 ON SELECTION BAR 或 ON SELECTION POPUP 命令可以执行一条命令。

菜单或菜单栏可以是用户自定义的（用 DEFINE POPUP 和 DEFINE MENU 创建），也可以是 Visual FoxPro 菜单系统的一部分。

示例

下面的示例演示了一个层叠子菜单系统。创建了有两个菜单栏标题的菜单栏 mnuDinner。每个标题用 ON PAD 激活 popMainCourse 或 popDessert 菜单。名称为 popMainCourse 和 popDessert 的菜单都有附加的菜单 popBurger、popPizza 和 popPie，它们是用三个 ON BAR 命令分配给项列表的。popOlives 和 popPie 项有用两个 ON BAR 命令分配的附加菜单。

作出选择后，ON SELECTION POPUP ALL 执行一个 yourchoice 过程，它激活窗口并且显示您的选择。用 POPUP( ) 和 PROMPT( ) 确定选择，并返回菜单名称和菜单项的内容（文本）。

```
DEFINE WINDOW wOrder FROM 10,0 TO 13,39
DEFINE MENU mnuDinner
DEFINE PAD padOne OF mnuDinner PROMPT '\<Main Course' KEY ALT+M, ''
DEFINE PAD padTwo OF mnuDinner PROMPT '\<Dessert'  KEY ALT+D, ''
ON PAD padOne OF mnuDinner ACTIVATE POPUP popMainCourse
```

```
ON PAD padTwo OF mnuDinner ACTIVATE POPUP dessert
DEFINE POPUP popMainCourse MARGIN MESSAGE ;
    'We have burgers and pizza today'
DEFINE BAR 1 OF popMainCourse PROMPT '\<Hamburgers'
DEFINE BAR 2 OF popMainCourse PROMPT '\<Pizza'
ON BAR 1 OF popMainCourse ACTIVATE POPUP burger
ON BAR 2 OF popMainCourse ACTIVATE POPUP pizza
DEFINE POPUP burger MARGIN MESSAGE ;
    'What would you like on your burger?'
DEFINE BAR 1 OF burger PROMPT '\<Ketchup'
DEFINE BAR 2 OF burger PROMPT '\<Mustard'
DEFINE BAR 3 OF burger PROMPT '\<Onions'
DEFINE BAR 4 OF burger PROMPT '\<Pickles'
DEFINE POPUP pizza MARGIN MESSAGE ;
    'Here are the available toppings'
DEFINE BAR 1 OF pizza PROMPT '\<Anchovies'
DEFINE BAR 2 OF pizza PROMPT '\<Green Peppers'
DEFINE BAR 3 OF pizza PROMPT '\<Olives'
DEFINE BAR 4 OF pizza PROMPT '\<Pepperoni'
ON BAR 3 OF pizza ACTIVATE POPUP olives
DEFINE POPUP olives MARGIN
DEFINE BAR 1 OF olives PROMPT '\<Black' MESSAGE 'Black olives?'
DEFINE BAR 2 OF olives PROMPT '\<Green' MESSAGE 'Green olives?'
DEFINE POPUP dessert MARGIN MESSAGE 'Our dessert offerings'
DEFINE BAR 1 OF dessert PROMPT '\<Brownies'
DEFINE BAR 2 OF dessert PROMPT '\<Cookies'
DEFINE BAR 3 OF dessert PROMPT '\<Ice Cream'
DEFINE BAR 4 OF dessert PROMPT '\<Pie'
ON BAR 4 OF dessert ACTIVATE POPUP pie
DEFINE POPUP pie MARGIN MESSAGE 'What kind of pie?'
```

```
DEFINE BAR 1 OF pie PROMPT '\<Blueberry'
DEFINE BAR 2 OF pie PROMPT '\<Cherry'
DEFINE BAR 3 OF pie PROMPT '\<Peach'
DEFINE BAR 4 OF pie PROMPT '\<Rhubarb'
ON SELECTION POPUP ALL DO yourchoice
ACTIVATE MENU mnuDinner
PROCEDURE yourchoice
ACTIVATE WINDOW wOrder
CLEAR
DO CASE
    CASE POPUP( ) = 'BURGER'
        @ 0,0 SAY 'A ' + POPUP( ) + ' order:'
        @ 1,0 SAY 'You ordered a burger with ' + LOWER(PROMPT( ))
    CASE POPUP( ) = 'PIZZA'
        @ 0,0 SAY 'A ' + POPUP( ) + ' order:'
        @ 1,0 SAY 'You ordered a pizza with ' + LOWER(PROMPT( ))
    CASE POPUP( ) = 'OLIVES'
        @ 0,0 SAY 'A ' + POPUP( ) + ' order:'
        @ 1,0 SAY 'You ordered a pizza with ' ;
            + LOWER(PROMPT( )) + ' olives'
    CASE POPUP( ) = 'DESSERT'
        @ 0,0 SAY 'A ' + POPUP( ) + ' order:'
        @ 1,0 SAY 'You ordered ' + LOWER(PROMPT( )) + ' for dessert'
    CASE POPUP( ) = 'PIE'
        @ 0,0 SAY 'A ' + POPUP( ) + ' order:'
        @ 1,0 SAY 'You ordered ' + LOWER(PROMPT( )) + ' pie'
ENDCASE
WAIT WINDOW
DEACTIVATE WINDOW wOrder
RETURN
```

请参阅

ACTIVATE MENU, DEFINE BAR, DEFINE MENU, DEFINE POPUP, ON SELECTION BAR, ON SELECTION POPUP

ON ERROR 命令

当出现错误时，指定要执行的命令。

语法

ON ERROR
[Command]

参数描述

Command

指定应执行的 Visual FoxPro **命令。执行此命令后，程序将从引起错误的程序行的下一行重新开始执行。但如果错误处理过程中包含** RETRY**，则重新执行引起错误的程序行。**

如果命令指定了错误出现时执行的一个过程，那么可以使用 ERROR( )**,** MESSAGE( )**,** LINENO( ) **和** PROGRAM( ) **将错误代码、错误消息、程序引号以及程序**

名称传递到此过程。这些信息可以用来纠正错误。

说明

若在程序运行时发生错误，Visual FoxPro 将执行 ON ERROR 中指定的命令。通常情况下，ON ERROR 使用 DO 来执行一个错误处理过程。

用不带命令的 ON ERROR 可以恢复默认的 Visual FoxPro 错误处理程序。

ERROR 过程不可嵌套。如果在 ON ERROR 过程中又发出了 ON ERROR 命令，则恢复默认的 Visual FoxPro 错误处理程序。

示例

```
ON ERROR DO errhand WITH ;
    ERROR( ), MESSAGE( ), MESSAGE(1), PROGRAM( ), LINENO( )
***下行可能产生错误***
USE nodatabase

ON ERROR  &&系统错误处理
PROCEDURE errhand
PARAMETER merror, mess, mess1, mprog, mlineno
CLEAR
? 'Error number: ' + LTRIM(STR(merror))
? 'Error message: ' + mess
? 'Line of code with error: ' + mess1
? 'Line number of error: ' + LTRIM(STR(mlineno))
? 'Program with error: ' + mprog
```

请参阅

AERROR( ), COMRETURNERROR( ), DO, ERROR( ), FUNCTION,

LINENO( ), MESSAGE( ), PROGRAM( ), PROCEDURE, RETRY

ON ESCAPE 命令

指定在程序或命令运行过程中，按下 ESC 键时所执行的命令。

语法

ON ESCAPE
[Command]

参数描述

Command

指定 Visual FoxPro 要执行的命令。命令执行之后，程序将从按下 ESC 键时所在执行程序行的下一行重新开始执行。但如果在 ON ESCAPE 指定的过程中包含 RETRY，则重新执行按下 ESC 键时正执行的程序行。

说明

一般来说，ON ESCAPE 使用 DO 来执行一个过程。

如果在按下 ESC 时，ON ESCAPE 和 ON KEY 都有效，Visual FoxPro 将执行 ON ESCAPE 指定的命令。

若使用不带命令的 ON ESCAPE，按下 ESC 键时将不执行任何命令(默认情况)。

注意 如果 SET ESCAPE 为 OFF，Visual FoxPro 将不执行 ON ESCAPE 过程。

示例

下面的示例设置了一个无限循环，但是定义了一个 ON ESCAPE 例程以退出。

```
SET ESCAPE ON

ON ESCAPE DO stopit
WAIT WINDOW 'Press ESC to stop loop' NOWAIT
glMoreLoop = .T.

DO WHILE glMoreLoop
ENDDO
RETURN

PROCEDURE stopit
glMoreLoop = .F.
RETURN
```

请参阅

INKEY( )

ON EXIT BAR 命令

包含此命令是为了提供与 dBASE 的兼容性。

ON EXIT MENU 命令

包含此命令是为了提供与 dBASE 的兼容性。

ON EXIT PAD 命令

包含此命令是为了提供与 dBASE 的兼容性。

ON EXIT POPUP 命令

包含此命令是为了提供与 dBASE 的兼容性。

ON KEY 命令

包含此命令是为了提供向后兼容性。请使用 ON KEY LABEL 命令。

ON KEY = 命令

包含此命令是为了提供向后兼容性。请使用 ON KEY LABEL 命令。

ON KEY LABEL 命令

指定当按下特定键或组合键，或者单击鼠标按钮时所要执行的命令。

语法

ON KEY [LABEL KeyLabelName] [Command]

参数描述

LABEL KeyLabelName

为键指定键标记，*KeyLabelName*是键上的字母或数字或者指定给该键的特殊名称。下表列出了特殊的键标记名称。

Visual FoxPro 键标记指定值：

键	键标记
←	LEFTARROW
→	RIGHTARROW
↑	UPARROW
↓	DNARROW
HOME	HOME
END	END

续 表

PAGE UP	PGUP
PAGE DOWN	PGDN
DEL	DEL
BACKSPACE	BACKSPACE
SPACEBAR	SPACEBAR
INS	INS
TAB	TAB
SHIFT+TAB	BACKTAB
ENTER	ENTER
F1 to F12	F1, F2, F3 ...
CTRL+F1 to CTRL+F12	CTRL+F1, CTRL+F2 ...
SHIFT+F1 to SHIFT+F12	SHIFT+F1, SHIFT+F2 ...
ALT+F1 to ALT+F12	ALT+F1, ALT+F2, ALT+F3 ...
ALT+0 to ALT+9	ALT+0, ALT+1, ALT+2 ...
ALT+A to ALT+Z	ALT+A, ALT+B, ALT+C ...
CTRL+LEFT ARROW	CTRL+LEFTARROW
CTRL+RIGHT ARROW	CTRL+RIGHTARROW
CTRL+HOME	CTRL+HOME
CTRL+END	CTRL+END
CTRL+PAGE UP	CTRL+PGUP

续表

CTRL+PAGE DOWN	CTRL+PGDN
CTRL+A TO CTRL+Z	CTRL+A, CTRL+B, CTRL+C ...
CTRL+0	CTRL+0
RIGHT MOUSE BUTTON	RIGHTMOUSE
LEFT MOUSE BUTTON	LEFTMOUSE
MOUSE BUTTON	MOUSE
ESC	ESC

Command

指定当按下特定键或组合键或者单击鼠标按钮时所执行的命令。

在分配给键的命令中也可以带参数或参数表达式，例如，

```
ON KEY LABEL ALT+V WAIT WINDOW "Version: " + VERSION()
```

在命令中可以包含变量，但是必须是公共的。例如，

```
PUBLIC message
message = "Default drive: " + SYS(5)
ON KEY LABEL ALT+D WAIT WINDOW message
```

说明

一般来说，ON KEY LABEL 使用 DO 来执行一个过程。

在 READ、BROWSE、EDIT、CHANGE 或者用户自定义菜单的执行过程中，ON

KEY LABEL 立即执行命令。如果按键或者单击鼠标时某程序正在运行，Visual FoxPro 将执行完该程序当前行后再执行 ON KEY LABEL 命令。在程序中创建的任何 ON KEY LABEL 键指定值在程序运行完毕后仍然有效。也可以在命令窗口中创建键指定值。

如果要恢复指定键的动作，可以使用 ON KEY LABEL *KeyLabelName* 命令。如果要把全部键的行为恢复正常，应使用 ON KEY 命令。

提示 为防止在执行 ON KEY LABEL 过程中递归调用，应在过程前面包含 PUSH KEY CLEAR 命令以禁止所有激活的 ON KEY LABEL 命令。然后在过程末尾发出一个 POP KEY 命令来启用 ON KEY LABEL 命令。

ON KEY LABEL 键指定值在 Visual FoxPro 系统菜单栏、系统菜单、对话框和警报器等处不起作用。它们只在 Visual FoxPro 系统窗口、Visual FoxPro 文本编辑器、命令窗口、跟踪窗口等处有效。

与 ON KEY 不同，可以有多个激活的 ON KEY LABEL 命令。例如，可以为每个箭头键和鼠标按钮指派命令。

注意在执行 ON KEY LABEL 时，PARAMETERS( ) 设置为 0。详细内容，请参阅稍后的“语言参考”中的 PARAMETERS( ) 函数。

Visual FoxPro 中某些事件不可被捕获，因为它们在相应窗口的控制之下。特别的，当单击一个控制菜单或滚动条之类的窗口控制时，不执行 ON KEY LABEL MOUSE，ON KEY LABEL LEFTMOUSE 和 ON KEY LABEL RIGHTMOUSE。还要注意在 Visual FoxPro 中，ON KEY LABEL 支持 CTRL+0，它将该组合键重定义为向字段中输入空值。

附注 ON KEY LABEL **的操作范围在表单之外；可以使用表单的** KeyPress **事件，当按下一个键时执行代码。**

示例

下面的示例在按下箭头键时，显示信息。

```
CLEAR
PUBLIC msg
msg = CHR(13) + CHR(13) + "Press F9 to " + ;
    restore default key definition."

ON KEY LABEL RIGHTARROW Wait Window "Right Arrow " + msg NOWAIT

ON KEY LABEL LEFTARROW Wait Window "Left Arrow " + msg NOWAIT

ON KEY LABEL UPARROW Wait Window "Up Arrow " + msg NOWAIT

ON KEY LABEL DNARROW Wait Window "Down Arrow " + msg NOWAIT

* Press F9 to clear the ON KEY LABEL assignments

ON KEY LABEL F9 ON KEY
```

请参阅

INKEY( ), KeyPress **事件**, ON( ), POP KEY, PUSH KEY

ON PAD 命令

指定菜单或菜单栏，当选择特定的菜单标题时激活它。

语法

ON PAD MenuTitleName OF MenuBarName1
[ACTIVATE POPUP MenuName
|ACTIVATE MENU MenuBarName2

参数描述

MenuTitleName OF MenuBarName1
指定一个具体的菜单标题。

ACTIVATE POPUP MenuName
指定选择菜单标题时激活的菜单。使用不带 ACTIVATE POPUP 子句的 ON PAD *MenuTitleName* OF *MenuBarName1* 命令，可以从菜单标题中释放一个菜单。

在 Visual FoxPro 中，如果为 *MenuBarName1* 指定了系统菜单栏 _MSYSMENU，则 *MenuName* 不能指定用 DEFINE POPUP PROMPT 子句创建的菜单。

ACTIVATE MENU MenuBarName2
指定选择菜单标题时激活的菜单栏名称。使用不带 ACTIVATE MENU 子句

的 ON PAD *MenuTitleName* OF *MenuBarName1* 命令，可以从菜单名中释放菜单栏。

说明

使用 ON SELECTION PAD，可以在选择一个菜单标题时执行一条命令。

请参阅

ACTIVATE MENU, DEFINE MENU, ON SELECTION MENU, ON SELECTION PAD

ON PAGE 命令

当报表中打印输出到达一定行，或发出 EJECT PAGE 时，将执行的命令。

语法

ON PAGE
[AT LINE nLineNumber [Command]]

参数描述

AT LINE nLineNumber [Command]

指定到达预定行数时执行的命令。当 _PLINENO（保存报表中当前行号的系统变量）

大于 *nLineNumber* 指定行号时，执行指定的命令。当发出 EJECT PAGE 命令时也执行 ON PAGE 指定的命令。详细内容，请参阅前面的“语言参考”中的 EJECT PAGE 命令。

说明

ON PAGE 通常使用 DO 来执行一个处理分页、标头和注脚的过程。

不带 AT LINE 子句执行 ON PAGE 命令，将清除 ON PAGE 命令。

请参阅

EJECT PAGE

ON READERROR 命令

包含向后兼容性。用 Valid 事件代替。

返回总目录

PAD（） 函数
PADL（） | PADR（） | PADC（） 函数
_PADVANCE 系统变量
Page 对象
PageCount 属性
PageFrame 控件
PageHeight 属性
_PAGENO 系统变量
PageOrder 属性
Pages 属性
PageWidth 属性
Paint 事件
Panel 属性
PanelLink 属性
PARAMETERS 命令
PARAMETERS（） 函数
Parent Object 参考
ParentAlias 属性
ParentClass 属性
Partition 属性
PasswordChar 属性

PAYMENT（） 函数
_PBPAGE 系统变量
PCOL（） 函数
_PCOLNO 系统变量
_PCOPIES 系统变量
PCOUNT（） 函数
_PDRIVER 系统变量
_PDSETUP 系统变量
_PECODE 系统变量
_PEJECT 系统变量
PEMSTATUS（） 函数
_PEPAGE 系统变量
PI（） 函数
Picture 属性
PLAY MACRO 命令
_PLENGTH 系统变量
_PLINENO 系统变量
_PLOFFSET 系统变量
Point 方法
POP KEY 命令
POP MENU 命令

ON SELECTION BAR 命令

指定选择特定菜单项时应执行的命令。

语法

ON SELECTION BAR nMenuItemNumber OF MenuName
[Command]

参数描述

nMenuItemNumber OF MenuName [Command]

指定菜单项的编号、菜单名称以及选定此菜单项时执行的命令。

说明

通常，ON SELECTION BAR 使用 DO 来执行一个过程或程序。当创建并激活一个菜单时，应把 ON SELECTION BAR 命令置于 DEFINE POPUP 和 ACTIVATE POPUP 之间。

可以对 Visual FoxPro 系统菜单使用 ON SELECTION BAR 命令。

使用 ON SELECTION BAR 命令，可以在选择菜单中任何菜单项时执行一条命令。使用 ON BAR 命令，可以在选择特定菜单项时激活一个菜单或菜单栏。

使用不带可选参数的 ON SELECTION BAR *nMenuItemNumber* OF *MenuName* 命令，可

以释放为菜单项指定的命令。

请参阅

DEFINE BAR, DEFINE POPUP, ON BAR, ON SELECTION POPUP

ON SELECTION MENU 命令

在菜单栏上选择任何菜单标题时，指定要执行的命令。

语法

ON SELECTION MENU MenuBarName | ALL
[Command]

参数描述

MenuBarName

要为其指定命令的菜单栏名称。从该菜单栏上选定任一菜单标题时都执行此命令。可以指定 DEFINE MENU 命令创建的用户自定义菜单名或 Visual FoxPro 系统菜单栏 _MSYSMENU。

ALL

从任何菜单栏上，选择任一菜单标题时都执行此命令。

Command

指定选择菜单标题时要执行的命令。使用不带命令的 ON SELECTION MENU 命令，可以释放为菜单栏指定的命令。

说明

创建并激活菜单时，应将 ON SELECTION MENU 命令放置在 DEFINE MENU 命令和 ACTIVATE MENU 命令之间。

使用 ON SELECTION PAD 命令，可以在选择特定菜单项时执行某个命令。ON SELECTION PAD 命令的优先级高于 ON SELECTION MENU 命令。使用 ON PAD 命令，可以在选定特定菜单标题时激活某一菜单或菜单栏。使用不带命令的 ON SELECTION MENU 命令，可以释放为某菜单栏指定的命令。

用不带命令的 ON SELECTION MENU 可以释放指定给菜单栏的命令。

示例

下面的示例中，当从 Visual FoxPro 系统菜单栏中选中一个菜单标题时，用 ON SELECTION MENU 执行过程。

用 SET SYSMENU SAVE 将当前的系统菜单栏保存到内存中，并且用 SET SYSMENU TO 删除所有的系统菜单标题。

用 DEFINE PAD 创建几个系统菜单标题。当选中了一个菜单标题时，执行用 ON SELECTION MENU 分配给菜单栏的 choice 过程。choice 显示所选的菜单标题和菜单栏名称。如果选择了“退出”菜单标题，则恢复原始的 Visual FoxPro 系统菜单。

```
*** 给程序命名为 ONMENU.PRG ***
CLEAR
```

```
SET SYSMENU SAVE
SET SYSMENU TO
DEFINE PAD padSys OF _MSYSMENU PROMPT '\<System' COLOR SCHEME 3 ;
    KEY ALT+S, ''
DEFINE PAD padEdit OF _MSYSMENU PROMPT '\<Edit' COLOR SCHEME 3 ;
    KEY ALT+E, ''
DEFINE PAD padRecord OF _MSYSMENU PROMPT '\<Record' COLOR SCHEME 3 ;
    KEY ALT+R, ''
DEFINE PAD padWindow OF _MSYSMENU PROMPT '\<Window' COLOR SCHEME 3 ;
    KEY ALT+W, ''
DEFINE PAD padReport OF _MSYSMENU PROMPT 'Re\<ports' COLOR SCHEME 3 KEY ALT+P, ''
DEFINE PAD padExit OF _MSYSMENU PROMPT 'E\<xit' COLOR SCHEME 3 ;
    KEY ALT+X, ''

ON SELECTION MENU _MSYSMENU ;
    DO choice IN onmenu WITH PAD ( ) , MENU ( )
PROCEDURE choice
PARAMETER gcPad, gcMenu
WAIT WINDOW 'You chose ' + gcPad + ;
    ' from menu ' + gcMenu NOWAIT
IF gcPad = 'PADEXIT'
    SET SYSMENU TO DEFAULT
ENDIF
```

请参阅

ACTIVATE MENU, DEFINE MENU, ON PAD, ON SELECTION PAD

ON SELECTION PAD 命令

在选择菜单栏上特定菜单标题时，指定要执行的命令。

语法

ON SELECTION PAD MenuTitleName OF MenuBarName
[Command]

参数描述

MenuTitleName OF MenuBarName
指定要执行的命令的菜单标题名称。

Command
指定选择特定菜单标题时要执行的 Visual FoxPro 命令。

说明

当选择某个菜单标题时，Visual FoxPro 将执行 ON SELECTION PAD 指定的命令。从菜单栏上选择特定的菜单标题时，ON SELECTION PAD 命令通常使用 DO 执行某个过程或程序。在创建并激活菜单栏时，应将 ON SELECTION PAD 命令放置在 DEFINE MENU 命令与 ACTIVATE MENU 命令之间。

使用 ON SELECTION MENU 命令，可以在菜单栏上选定任一菜单标题时执行某个命令。使用 ON PAD 命令，可以在菜单栏上选择特定菜单标题时激活某个菜单或菜单

栏。

使用不带命令的 ON SELECTION PAD 命令，可以释放为菜单标题指定的命令。

请参阅

ACTIVATE MENU, DEFINE MENU, DEFINE PAD, ON PAD, ON SELECTION MENU

ON SELECTION POPUP 命令

从特定菜单或所有菜单上选择任一菜单项时，指定所要执行的命令。

语法

ON SELECTION POPUP MenuName | ALL
[Command]

参数描述

MenuName

指定要为其指定命令的菜单。

ALL

如果包含 ALL 而不是一个菜单名，则从任何菜单上选择任一菜单项时，Visual FoxPro 都执行此命令。

Command

指定选择菜单项时所要执行的命令。

说明

从菜单上选择任一菜单项时，Visual FoxPro 执行由 ON SELECTION POPUP 命令指定的命令。在创建并激活菜单时，应将 ON SELECTION POPUP 命令放置在 DEFINE POPUP 命令与 ACTIVATE POPUP 命令之间。

ON SELECTION BAR 命令的优先级高于 ON SELECTION POPUP 命令。使用 ON BAR 命令，可以在选择特定的菜单项时激活某个菜单或菜单栏。使用不带命令的 ON SELECTION POPUP 命令，可以释放 ON SELECTION POPUP 命令先前指定给菜单项的命令。

ON SELECTION BAR 命令的优先级高于 ON SELECTION POPUP 命令。使用 ON BAR 命令，可以在选择特定的菜单项时激活某个菜单或菜单栏。使用不带命令的 ON SELECTION POPUP 命令，可以释放 ON SELECTION POPUP 命令先前指定给菜单项的命令。

请参阅

ACTIVATE POPUP, DEFINE BAR, DEFINE POPUP, ON BAR

ON SHUTDOWN 命令

指定当试图退出 Visual FoxPro 或 Microsoft Windows 时所要执行的命令。

语法

ON SHUTDOWN [Command]

参数描述

不带 *Command* 发出 ON SHUTDOWN 命令，将释放当前的 ON SHUTDOWN 命令。

说明

当您试图退出 Visual FoxPro 时，将执行 ON SHUTDOWN 中指定的命令。如果在 Visual FoxPro 打开的情况下，试图从“程序管理器”退出 Microsoft Windows，控制权将返回给 Visual FoxPro 并执行 ON SHUTDOWN 中指定的命令。

ON SHUTDOWN 命令一般用 DO 命令执行某个例程。该例程显示一对话框，询问是否确定要退出当前应用程序和 Visual FoxPro。如果您要退出应用程序，该例程将关闭打开的文件并清理 Visual FoxPro 环境，然后执行 QUIT 命令。如果您不想退出当前的应用程序，例程将把控制权返回给应用程序。

请参阅

QUIT

ON（）函数

返回为下列事件处理命令指定的命令：ON ERROR、ON ESCAPE、ON KEY LABEL **或** ON PAGE。

语法

ON(cONCommand [, KeyLabelName])

返值类型

字符型

参数描述

cONCommand

指定其中的一个事件处理命令。下面是各个命令及在 ON（） **函数中使用的相应的字符表达式：**

命令	cONCommand
ON ERROR	ERROR
ON ESCAPE	ESCAPE
ON KEY LABEL	KEY

ON PAGE PAGE

例如，要返回当前为 ON ERROR 指定的命令，应使用：

```
? ON('ERROR')
```

KeyLabelName

用在 ON KEY LABEL 命令中，为命令指定键或组合键。把 *cONCommand* 指定为 KEY 后，在 *KeyLabelName* 中指定键或组合键的键标记名称。有关键标记名称的完整列表，请参阅 ON KEY LABEL。

例如，若要返回当前 ON KEY LABEL 命令为功能键 F7 指定的命令，应使用：

```
? ON('KEY', 'F7')
```

说明

当某一事件发生并被一个事件处理命令捕捉时，执行该事件处理命令指定的命令。ON（）函数返回为某个事件处理命令指定的命令。如果当前没有命令指定给该事件处理命令，ON（）函数返回空字符串。

示例

下面的示例用 ON（）显示 ON ERROR 和 ON KEY LABEL 设置。

```
ON ERROR DO errorhand
ON KEY LABEL CTRL+F2 WAIT WINDOW 'You pressed ^F2'
ON KEY LABEL ALT+Z DISPLAY MEMORY
CLEAR
? ON('ERROR') && 显示执行错误处理程序
```

```
? ON('KEY', 'CTRL+F2') && 显示 WAIT WINDOW 'You pressed ^F2'
? ON('KEY', 'ALT+Z') && 显示 DISPLAY MEMORY
ON ERROR
ON KEY LABEL CTRL+F2
ON KEY LABEL ALT+Z
```

请参阅

INKEY（）,LASTKEY（）,ON ERROR,ON ESCAPE,ON KEY,ON KEY LABEL,ONPAGE ,ON READERROR,READKEY（）

OneToMany 属性

当在父表的记录上移动记录指针时，指定记录指针是否应保持在同一条父记录上，直至子表的记录指针移过所有与之相关的记录。

语法

Object.DataEnvironment.Relation.OneToMany[= lExpr]

参数描述

lExpr

OneToMany **属性的设置有：**

设置	说明
“真”(.T.)	记录指针保持在同一条父记录上，直至子表的记录指针移过所有相关记录。
“假”(.F.)	（默认值）父表的记录指针移动到某一指定记录，子表的记录指针移动到第一条相关记录。

说明

OneToMany 属性可以指定简单的父-子表关系下记录指针的移动方式。要指定更复杂的关系(如祖父表-父表-子表关系)下记录指针的移动方式，可以在表单或报表的 Init 方法中使用 SET SKIP 命令。

OneToMany 属性设置为“真”(.T.)时，与 SET SKIP 命令的行为相似。

应用于

关系

请参阅

SET SKIP

OPEN DATABASE 命令

打开一个数据库

语法

OPEN DATABASE [FileName | ?]
[EXCLUSIVE | SHARED]
[NOUPDATE]
[VALIDATE]

参数描述

FileName

指定要打开的数据库的名称。如果没有指定该文件的扩展名，Visual FoxPro 自动指定为 .DBC。如果省略 *FileName* 将显示“打开”对话框。

注意 磁盘或目录名称含有感叹号(!)时，Pro 不能正确识别路径名。

?

显示“打开”对话框，从中可以选择现有的数据库，或输入所要创建的新表单的名称。

EXCLUSIVE

以独占方式打开数据库。如果以独占方式打开数据库，则其他用户无法访问该数据库，并且当他们试图访问时会产生错误。如果没有包含 EXCLUSIVE

和 SHARED，则当前 SET EXCLUSIVE 的设置值决定数据库以何种方式打开。

SHARED
以共享方式打开数据库。如果以共享方式打开数据库，其他的用户也可以访问它。如果没有包含 EXCLUSIVE 和 SHARED，则当前 SET EXCLUSIVE 的设置值决定数据库以何种方式打开。

NOUPDATE
指定不能对数据库做任何更改。换句话说，该数据库只读。如果省略 NOUPDATE，则数据库打开后可以进行读写。

数据库中包含的表不受 NOUPDATE 的影响。要防止对数据库中某个表的更改，打开该表时，应在 USE 命令中包含 NOUPDATE。

VALIDATE
指定让 Visual FoxPro 确保数据库中的引用有效。Visual FoxPro 将检查磁盘上数据库中的表和索引是否可用。Visual FoxPro 还将检查被引用的字段和索引标识是否存于表和索引中。

说明

数据库打开时，其中包含的所有表均可用。不过，表不能隐含地打开。必须用 USE 命令打开表。
执行 USE 命令时，Visual FoxPro 首先在当前数据库中查找该表。如果没有找到，

Visual FoxPro 接着在此数据库之外查找该表。这意味着如果数据库中的某个表与数据库之外的一个表具有相同的名称，则首先找到的是数据库中的表。

您不能打开其他用户以独占方式打开的数据库。

示例

在下面的 example 中，使用 OPEN DATABASE 命令打开数据库 testdata。然后用 DISPLAY DATABASE 命令显示数据库中各个表的有关信息。

```
CLOSE DATABASES
SET PATH TO (HOME(2) + 'Data\')    && 设置数据库路径

OPEN DATABASE testdata  && 打开 testdata 数据库
DISPLAY DATABASE  && 显示表信息
```

请参阅

ADD TABLE, CLOSE DATABASES, CREATE DATABASE, DBUSED（）, DISPLAY DATABASE , FREE TABLE, LIST DATABASE, REMOVE TABLE

OpenTables 方法

以编程方式打开与数据库环境相关的表或视图。

语法

DataEnvironment.OpenTables

参数描述

DataEnvironment

指定与表单集、表单或报表相关的数据环境。

说明

当数据环境的 AutoOpenTables 属性设置为“假”(.F.)，或已经用 CloseTables 方法卸载数据环境后，可以用 OpenTables 方法加载数据环境的表。

应用于

数据环境

请参阅

AfterCloseTables 事件, AutoOpenTables 属性, BeforeOpenTables 事件, CloseTables 方法

OpenViews 属性

确定与表单集、表单或报表的数据环境相关的视图的类型，该视图自动打开。设计时可用；运行时只读。

语法

DataEnvironment.OpenViews[= nExpression]

参数描述

nExpression

OpenViews 属性的设置有：

设置	说明
0	（默认值）本地和远程。表单集、表单或报表的数据环境的本地和远程视图都自动打开。
1	本地。只有表单集、表单或报表的数据环境的本地视图自动打开。
2	远程。只有表单集、表单或报表的数据环境的远程视图自动打开。
3	无。对于表单集、表单或报表的数据环境，不打开视图。

说明

当 AutoOpenTables 属性设置为“真”(.T.)，或者执行 OpenTables 方法时，可以使用 OpenViews 属性。

应用于

数据环境

请参阅

AutoOpenTables 属性, OpenTables 方法

OpenWindow 属性

包含此属性是为了向后兼容性。可用编辑框控件代替。

OptionButton 控件

创建单个选项按钮。

语法

OptionButton

说明

只能在选项按钮组中添加单个选项按钮。

有关创建选项按钮组的详细内容，请参阅稍后的“OptionGroup Control”和《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十章“使用控件”。

属 性

Alignment
Application
AutoSize
BackColor
BackStyle
BaseClass
Caption
Class
ClassLibrary
ColorScheme
ColorSource
Comment
ControlSource
DisabledBackColor
DisabledForeColor
DisabledPicture
DownPicture
DragIcon
DragMode
Enabled
FontBold
FontCondense
FontExtend
FontItalic
FontName
FontOutline
FontShadow
FontSize
FontStrikeThru
FontUnderLine
ForeColor
Height
HelpContextID
Left
MousePointer
Name
OLEDragMode
OLEDragPicture
OLEDropEffects
OLEDropHasData
OLEDropMode
Parent
ParentClass
Picture
RightToLeft
SpecialEffect
StatusBarText
Style
TabIndex
TabStop
Tag
TerminateRead
ToolTipText
Top

续 表

Value	Visible	WhatsThisHelpID
Width		

事 件

Click	DblClick	Destroy
DragDrop	DragOver	Error
ErrorMessage	GotFocus	Init
KeyPress	LostFocus	Message
MiddleClick Event	MouseDown	MouseMove
MouseUp	MouseWheel	OLECompleteDrag
OLEDragDrop	OLEDragOver	OLEGiveFeedBack
OLESetData	OLEStartDrag	RightClick
Valid	When	

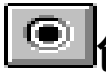
方 法

AddProperty	CloneObject	Drag
Move	OLEDrag	ReadExpression
ReadMethod	Refresh	ResetToDefault
SaveAsClass	SetFocus	WriteExpression
WriteMethod	Zorder	

请参阅

CREATE FORM, CREATE CLASS, DEFINE CLASS, OptionGroup 控件

OptionGroup 控件

 创建一组选项按钮。

语法

OptionGroup

说明

选项按钮组是包含选项按钮的容器。选项按钮组允许您从中选择一个按钮。选定某个选项按钮将释放先前的选择同时使您的选择成为当前值。选项按钮旁边的圆点指示当前的选择。例如，可以用选项按钮决定输出到文件、打印机还是窗口。

有关创建选项按钮组的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十章“使用控件”。

属 性

Application
AutoSize
BackColor
BackStyle
BaseClass
BorderColor
BorderStyle
ButtonCount
Buttons
Class
ClassLibrary
Comment
ColorSource
ControlSource
DragIcon
DragMode
Enabled
Height
HelpContextID
Left
MouseIcon
MousePointer
Name
OLEDragMode
OLEDragPicture
OLEDropEffects
OLEDropHasData
OLEDropMode
Parent
ParentClass
SpecialEffect
TabIndex
Tag
TerminateRead
Top
Value
Visible
WhatsThisHelpID
Width

事 件

Click
DblClick
DragDrop
DragOver
Error
ErrorMessage
Init
InteractiveChange
Message
MiddleClick
MouseDown
MouseMove

续表

MouseUp　MouseWheel　OLECompleteDrag
OLEDragDrop　OLEDragOver　OLEGiveFeedBack
OLESetData　OLEStartDrag　ProgrammaticChange
RightClick　UIEnable　Valid
When

方法

AddObject　AddProperty　CloneObject
Drag　Move　OLEDrag
ReadExpression　ReadMethod　Refresh
RemoveObject　ResetToDefault　SaveAsClass
SetAll　WriteExpression　WriteMethod
ZOrder

示例

下面的示例创建了一个选项组控件，并放置在表单上。选项组控件有三个按钮，单击不同的选项按钮，会显示圆、椭圆或长方形。Buttons 和 Caption 属性用来指定每个选项按钮边显示的文本。
用形状控件来创建圆、椭圆和长方形。单击一个选项按钮时，选项组控件的 Click 事件用 DO CASE ... ENDCASE 结构和 Value 属性来显示合适的形状。

```
frmMyForm = CREATEOBJECT('Form') && 创建一个表单
frmMyForm.Closable = .F. && 使控件菜单栏不可用

frmMyForm.AddObject('cmdCommand1','cmdMyCmndBtn') && 增加命令按扭
frmMyForm.AddObject('opgOptionGroup1','opgMyOptGrp') && 增加选项组
frmMyForm.AddObject('shpCircle1','shpMyCircle') && 增加圆形
frmMyForm.AddObject('shpEllipse1','shpMyEllipse') && 增加椭圆形
frmMyForm.AddObject('shpSquare','shpMySquare') && 增加 BOX形

frmMyForm.cmdCommand1.Visible =.T. && "退出"命令按扭可见

frmMyForm.opgOptionGroup1.Buttons(1).Caption = "\<Circle"
frmMyForm.opgOptionGroup1.Buttons(2).Caption = "\<Ellipse"
frmMyForm.opgOptionGroup1.Buttons(3).Caption = "\<Square"
frmMyForm.opgOptionGroup1.SetAll("Width", 100) && 设置选项组宽
frmMyForm.opgOptionGroup1.Visible = .T. && 选项组可见
frmMyForm.opgOptionGroup1.Click && 显示圆形

frmMyForm.SHOW && 显示表单
READ EVENTS && 启动事件程序

DEFINE CLASS opgMyOptGrp AS OptionGroup && 创建选项组
    ButtonCount = 3 && 三个选项按扭
    Top = 10
    Left = 10
    Height = 75
    Width = 100

    PROCEDURE Click
        ThisForm.shpCircle1.Visible = .F. && 隐藏圆形
```

```
        ThisForm.shpEllipse1.Visible = .F.  && 隐藏椭圆形
        ThisForm.shpSquare.Visible = .F.  && 隐藏方形

        DO CASE
            CASE ThisForm.opgOptionGroup1.Value = 1
                ThisForm.shpCircle1.Visible = .T. && 显示圆形
            CASE ThisForm.opgOptionGroup1.Value = 2
                ThisForm.shpEllipse1.Visible = .T.  && 显示椭圆形
            CASE ThisForm.opgOptionGroup1.Value = 3
                ThisForm.shpSquare.Visible = .T.  && 显示方形
        ENDCASE
ENDDEFINE

DEFINE CLASS cmdMyCmndBtn AS CommandButton  && 创建命令按扭
    Caption = '\<Quit'  && 给命令按扭增加说明
    Cancel = .T.  && 默认取消命令按扭(Esc键)
    Left = 125  && 命令按扭列
    Top = 210  && 命令按扭行
    Height = 25  && 命令按扭高
    PROCEDURE Click
        CLEAR EVENT  && 终止事件程序，关闭表单
ENDDEFINE

DEFINE CLASS shpMyCircle AS SHAPE  && 画一个圆
    Top = 10
    Left = 200
    Width = 100
    Height = 100
    Curvature = 99
    BackColor = RGB(255,0,0)  && 红色
```

```
ENDDEFINE

DEFINE CLASS shpMyEllipse AS SHAPE  && 画一个椭圆
    Top = 35
    Left = 200
    Width = 100
    Height = 50
    Curvature = 99
    BackColor = RGB(0,128,0)  && 绿色
ENDDEFINE

DEFINE CLASS shpMySquare AS SHAPE  && 画一个方形
    Top = 10
    Left = 200
    Width = 100
    Height = 100
    Curvature = 0
    BackColor = RGB(0,0,255)  && 蓝色
ENDDEFINE
```

请参阅

CREATE FORM, CREATE CLASS, DEFINE CLASS, OptionButton 控件

Order 属性

为临时表对象指定主控索引标识。设计和运行时可用。

语法

DataEnvironment.Cursor.Order[= cTagName]

参数描述

cTagName

指定存在于某个表中的索引标识字段。

说明

如果关系对象的 ChildOrder 属性已经设置，则忽略 Order 属性。用 Order 属性来指定记录显示或访问的顺序。设置 Order 属性更改索引顺序，并且移动记录指针到临时表中的第一个记录。

应用于

临时表

请参阅

ChildOrder 属性, INDEX, SET ORDER

ORDER（） 函数

返回当前表或指定表的主控索引文件或标识。

语法

ORDER([nWorkArea | cTableAlias [, nPath]])

返值类型

字符型

参数描述

nWorkArea

指定表所在的工作区，ORDER（） 函数返回该表的主控索引文件或标识名称。

cTableAlias

指定表的别名，ORDER（） 函数返回该表的主控索引文件或主控标识名称。

nPath

指定与单项索引文件名或复合索引文件名一起返回其所在驱动器及目录。数值表达式 *nPath* 可取任何值。

说明

一个表可以同时打开多个索引文件。不过，只有一个单项索引文件(主控索引文件)或复合索引文件中的标识(主控标识)控制着显示或访问表的顺序。有些命令，例如 SEEK 命令，它使用主控索引文件或主控标识搜索记录。本函数用来返回主控索引文件或主控标识的名称。

USE 与 SET INDEX 命令都支持用索引文件列表打开一个以上的索引。在这个索引文件列表中可以指定主控索引文件或主控标识。另外也可以用 SET ORDER 命令指定主控索引或主控标识。

默认情况下，ORDER（）函数返回当前工作区的主控索引文件名或主控标识名。如果没有设置顺序（发出 SET ORDER TO 命令或不存在主控索引文件和标识），则 ORDER（）函数返回空字符串。

示例

下面的 example 显示索引标识及文件。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
USE customer ORDER cust_id   && 打开 Customer 表
? ORDER（）   && display CUST_ID
? ORDER('customer', 1)   && 显示 CUSTOMER.CDX
```

请参阅

INDEX, SET INDEX, SET ORDER, USE

OS（） 函数

返回运行当前 Visual FoxPro 操作系统的名称和版本号。

语法

OS([1 | 2])

返值类型

字符型

参数描述

1

指定了返回的操作系统名称和版本号。

2

指定了一个表示操作系统是否支持返回 DBCS（双字节字符集）的字符串。如果支持 DBCS，则返回字符串“DBCS”，否则返回空字符串。

说明

如果省略了可选的参数 1 或 2，则返回下面的操作系统名称和版本号。

示例

```
? OS（）
? OS(1)
```

? OS(2)

请参阅

DISKSPACE（），GETENV（），VERSION（）

PACK 命令

从当前表中永久删除标有删除标记的记录，减少与该表相关的备注文件所占用的空间。

语法

PACK [MEMO] [DBF]

参数描述

MEMO

从备注文件中删除未使用空间，但不从表中删除标有删除标记的记录。备注字段的信息保存在一个相关的备注文件内。备注文件的文件名与表相同，扩展名为 .FPT。

DBF

从表中删除标有删除标记的记录，但不影响备注文件。

说明

当使用 PACK 命令时，Visual FoxPro 把所有没做删除标记的记录复制到一个临时表 (temporary table) 中。执行完 PACK 命令后，Visual FoxPro 把原表从磁盘上删除，同时用原表名命名临时表。如果按 ESC 键中止 PACK，就会删除临时表，原表保持不变。运行 PACK 命令时，如果磁盘空间不够，原表也将保持不变。

如果不带 MEMO 和 DBF 子句发出 PACK 命令，PACK 命令将同时作用于表和备注文件。

如果不带 MEMO 和 DBF 子句发出 PACK 命令，PACK 命令将同时作用于表和备注文件。

PACK 命令需要以独占方式使用表，有关在网络上以独占方式打开一个表的详细内容，请参阅 SET EXCLUSIVE。

如果当前表有一个或更多打开的索引，PACK 命令将重建索引文件。

警告 应仅在不再使用的记录上作标记。使用 PACK 命令之后，不可能再恢复已删除的记录。

请参阅

DELETE – SQL，DELETE，DELETED（），RECALL，SET EXCLUSIVE

PACK DATABASE 命令

从当前数据库中删除标有删除标记的记录。

语法

PACK DATABASE

说明

当从数据库中删除一个表或视图，或者修改数据库中一个表的结构时，数据库中就会包含做过删除标记的记录。

必须以独占方式打开数据库，并且数据库中不能打开任何表或视图。

示例

在下面的 example 中，使用 PACK DATABASE 命令清理 testdata 数据库，删除标有删除标记的记录。

```
CLOSE DATABASES
SET PATH TO (HOME(2) + 'data\')  && 设置数据库的路径
OPEN DATABASE testdata  && 打开数据库
PACK DATABASE && 清理当前数据库
```

请参阅

CLOSE DATABASES, DELETE DATABASE, MODIFY DATABASE,

PAD（） 函数

以大写字符串形式返回在某菜单栏中最近选取的菜单标题，或者返回一个逻辑值，表明一个活动菜单栏是否定义了菜单标题。

语法

PAD([cMenuTitle [, cMenuBarName]])

返值类型

字符型或逻辑值型

参数描述

cMenuTitle

指定菜单栏中菜单标题的名称。包含这个参数，可以检测一个活动菜单栏是否定义了菜单标题。如果定义了菜单标题，则返回“真”(.T.)，否则返回“假”(.F.)。

cMenuBarName

指定菜单栏的名称，该菜单栏包含菜单标题 cMenuTitle。如果省略 cMenuBarName，则假设菜单标题位于当前活动菜单栏中。

说明

为使 PAD（）函数返回一个菜单标题，必须定义且激活一个菜单栏。可以使用 DEFINE MENU 和 ACTIVATE MENU 命令创建和激活菜单栏。

PAD（）函数也可以应用于 Visual FoxPro 的系统菜单栏。

如果没有定义并激活一个菜单栏或者在命令窗口中使用 PAD（）函数，PAD（）（不带任何可选参数）函数将返回一个空字符串。

示例

本示例用 PAD（）将菜单标题传递到过程中。

用 SET SYSMENU SAVE 将当前的 Visual FoxPro 系统菜单栏保存到内存中，并且用 SET SYSMENU TO 删除所有的系统菜单标题。

用 DEFINE PAD 创建几个系统菜单标题。当选中了一个菜单标题时，用 PAD（）将菜单标题传递到过程 choice 中。choice 显示所选的菜单标题和菜单栏名称。如果选择了“退出”菜单标题，则恢复原始的 Visual FoxPro 系统菜单。

*** 给程序命名为 PADEXAM.PRG ***

```
CLEAR
SET SYSMENU SAVE
SET SYSMENU TO
DEFINE PAD padSys OF _MSYSMENU PROMPT '\<System' COLOR SCHEME 3 ;
    KEY ALT+S, ''
DEFINE PAD padEdit OF _MSYSMENU PROMPT '\<Edit' COLOR SCHEME 3 ;
    KEY ALT+E, ''
DEFINE PAD padRecord OF _MSYSMENU PROMPT '\<Record' COLOR SCHEME 3 ;
    KEY ALT+R, ''
DEFINE PAD padWindow OF _MSYSMENU PROMPT '\<Window' COLOR SCHEME 3 ;
```

```
    KEY ALT+W, ''
DEFINE PAD padReport OF _MSYSMENU PROMPT 'Re\<ports' COLOR SCHEME 3 KEY ALT+P, ''
DEFINE PAD padExit OF _MSYSMENU PROMPT 'E\<xit' COLOR SCHEME 3 ;
    KEY ALT+X, ''
ON SELECTION MENU _MSYSMENU ;
    DO choice IN padexam WITH PAD（） , MENU（）
PROCEDURE choice
PARAMETERS gcPad, gcMenu
WAIT WINDOW 'You chose ' + gcPad + ;
    ' from menu ' + gcMenu NOWAIT
IF gcPad = 'PADEXIT'
    SET SYSMENU TO DEFAULT
ENDIF
```

请参阅

ACTIVATE MENU, BAR（）, DEFINE PAD, DEFINE MENU, MENU（）, ON PAD, ON SELECTION PAD, PROMPT （）

PADL（） | PADR（） | PADC（） 函数

由表达式返回一个字符串，并从左边、右边或同时从两边用空格或字符把该字符串填充到指定长度。

语法

PADL(eExpression, nResultSize [, cPadCharacter])

–**或者** –

PADR(eExpression, nResultSize [, cPadCharacter])

–**或者** –

PADC(eExpression, nResultSize [, cPadCharacter])

返值类型

字符型

参数描述

eExpression

指定需填充的表达式。除了逻辑或货币表达式、通用字段或图片字段，这个表达式可以是任何类型的表达式。

nResultSize

指定表达式填充之后的字符总数。

cPadCharacter

指定用于填充的值。必要时这个值重复填充在表达式中，以使字符达到特定数目。

如果省略 cPadCharacter，**则使用空格（**ASCII **码为** 32 **填充）。**

说明

PADL（）**函数从左边插入填充值，**PADR（）**函数从右边插入填充值，**PADC（）

函数从两边插入填充值。

示例

```
STORE 'TITLE' TO gcString
CLEAR
? PADL(gcString, 40, '=')
? PADR(gcString, 40, '=')
? PADC(gcString, 40, '=')
```

请参阅

ALLTRIM（），LEFT（），LTRIM（），RIGHT（），STUFF（），TRIM（）

_PADVANCE 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

Page 对象

在页框中创建一个页面。

语法

Page

说明

使用页面可以创建带选项卡的表单或对话框。一组页面包含在页框中。

可以象下例这样，通过名称引用页框中的一个页面：

```
myFrame.MyPage1
```

也可以使用 PAGES 关键字通过页面的编号引用该页面。这与 Visual FoxPro 其他容器中的控件集合相一致。

```
myFrame.PAGES(2).Enabled = .T.
```

这个索引不必等于 the PageOrder 属性。如果有三个页面，它们的 PageOrders 分别是 2、3 和 5。就可以这样引用这三个页面：

```
myFrame.PAGES(1)
myFrame.PAGES(2)
myFrame.PAGES(3)
```

当具有页面的表单发生 Refresh 方法时，只刷新活动页面。

有关创建页面和页框的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十章“使用控件”中的“扩展表单”。

属性

ActiveControl	Application	BackColor
BackStyle	BaseClass	Caption
Class	ClassLibrary	ColorSource
Comment	ControlCount	Controls
DragIcon	DragMode	Enabled
FontBold	FontCondense	FontExtend
FontItalic	FontName	FontOutline
FontShadow	FontSize	FontStrikeThru
FontUnderline	ForeColor	HelpContextID
MouseIcon	MousePointer	Name
OLEDragMode	OLEDragPicture	OLEDropEffects
OLEDropHasData	OLEDropMode	PageOrder
Parent	ParentClass	Picture
Tag	ToolTipText	WhatsThisHelpID

事件

Activate	Click	DblClick

续 表

Deactivate	Destroy	DragDrop
DragOver	Error	Init
MiddleClick	MouseDown	MouseMove
MouseUp	MouseWheel	OLECompleteDrag
OLEDragDrop	OLEDragOver	OLEGiveFeedBack
OLESetData	OLEStartDrag	RightClick

方 法

AddObject	AddProperty	CloneObject
Drag	OLEDrag	ReadExpression
ReadMethod	Refresh	RemoveObject
ResetToDefault	SaveAsClass	SetAll
WriteExpression	WriteMethod	Zorder

请 参 阅

CREATE FORM, CREATE CLASS, DEFINE CLASS, PageFrame **控 件**

PageCount 属性

指定一个页框控件中的页面数。

语法

PageFrame.PageCount[= nPages]

参数描述

nPages

指定一个页框对象中包含的页面数。

说明

PageCount 属性的最小设置值为 0，最大值为 99。

注意

如果减少 PageCount 属性设置值（例如，由 3 修改为 2），将丢失所有超出新设置值的页面及它们包含的对象。

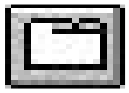
应用于

页框

请参阅

ButtonCount 属性, ControlCount 属性, FormCount 属性, Page 对象

PageFrame 控件



创建一个用以包含页面的页框。

语法

PageFrame

说明

页框是包含页面的容器对象，而页面可以包含控件。注意，要使页框可见，必须把它添加到一个表单中。

页框定义了页面的总体特性：大小和位置，边框类型，哪个页面是活动的等等。

页框决定了页面的位置及每页中有多少是可见的。页面相对于页框的左上角定位。如果页框移动了，页面跟着移动。

页框包含的每个页面默认命名为 Page1，Page2，Page3 等等。

注意 对页面所在的表单使用 Refresh 方法时，只刷新活动的页面。

有关创建页面和页框的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员

指南》的第十章“使用控件”中的“扩展表单”。

属性

ActivePage
Application
BaseClass
BorderColor
BorderWidth
Class
ClassLibrary
ColorSource
Comment
DragIcon
DragMode
Enabled
HelpContextID
Height
Left
MouseIcon
MousePointer
Name
OLEDragMode
OLEDragPicture
OLEDropEffects
OLEDropHasData
OLEDropMode
PageCount
PageHeight
Pages
PageWidth
Parent
ParentClass
RightToLeft
SpecialEffect
TabIndex
Tabs
TabStop
TabStretch
TabStyle
Tag
Top
Visible
WhatsThisHelpID
Width

事件

Click
DblClick
Destroy
DragDrop
DragOver
Error
Init
MiddleClick
MouseDown

续 表

MouseMove	MouseUp	MouseWheel
Moved	OLECompleteDrag	OLEDragDrop
OLEDragOver	OLEGiveFeedBack	OLESetData
OLEStartDrag	Resize	RightClick
UIEnable		

方 法

AddObject	AddProperty	CloneObject
Drag	Move	OLEDrag
ReadExpression	ReadMethod	Refresh
RemoveObject	ResetToDefault	SaveAsClass
SetAll	WriteExpression	WriteMethod
ZOrder		

请 参 阅

CREATE FORM, CREATE CLASS, DEFINE CLASS, Page **对 象**

PageHeight 属性

指定页面的高度。设计和运行时只读。

语法

PageFrame.PageHeight[= nHeight]

参数描述

nHeight

按照表单 ScaleMode 属性指定的度量单位指定页面的高度。

应用于

页框

请参阅

PageWidth 属性, Page 对象, ScaleMode 属性

_PAGENO 系统变量

包含当前页码。

语法

_PAGENO = nCurrentPageNumber

参数描述

nCurrentPageNumber

为当前页码指定一个从 1 到 32,767 之间的数值。_PAGENO，_PBPAGE 和 _PEPAGE 一起使用。如果设置（或增加）PAGENO，使之超出了 _PBPAGE 和 _PEPAGE 的范围，则不打印任何页面。

说明

_ PAGENO 包含了一个决定当前页码的数值，初始值默认为 1。_PAGENO 允许在连续输出中打印页码，而不必为此定义、初始化和增加一个变量。

请参阅

ON PAGE, _PBPAGE, _PEPAGE, _PLENGTH, _PLINENO

PageOrder 属性

指定页面在一个页框控件中的相对顺序。设计和运行时可用。

语法

Page.PageOrder[= nOrder]

参数描述

nOrder

指定在一个页框对象中页面的相对顺序。

说明

如果一个页框包含五个页面，且需要将第三页放在最后，则把第三页的 PageOrder 属性设置为 5。第四页的 PageOrder 设置为 3，第五页的 PageOrder 设置为 4，依此类推。

注意 PageOrder 的设置不一定是连续，可以有间隔。例如，如果一个页框包含三个页面，可以再增加第四个页面，并且将它的 PageOrder 属性设置为 10。

应用于

页面

请参阅

Activate 事件, Deactivate 事件, PageFrame 控件, ZOrder 方法

Pages 属性

一个用于访问页框控件中各个页面的数组。运行时可用。

语法

PageFrame.Pages(Index).Property[= Expr]

参数描述

Index

代表要访问页面的索引。

Expr

为页框中包含的页面指定一个属性值。

应用于

页框

请参阅

Buttons 属性, Page 对象, PageCount 属性

PageWidth 属性

指定页面的宽度。设计和运行时可用。

语法

PageFrame.PageWidth[= nWidth]

参数描述

nWidth

按照表单的 ScaleMode 属性指定的度量单位指定页面的宽度。

说明

除非 PageFrame 对象包含不止一个页面，并且页面是层叠的，否则 PageWidth 属性的设置与 Width 属性的设置总是相同的。

应用于

页框

请参阅

PageHeight 属性, Page 对象, ScaleMode 属性

Paint 事件

当表单或工具栏重画时发生。

语法

PROCEDURE Object.Paint

说明

当表单或工具栏移动或调整大小，或一个覆盖表单或工具栏的窗口移动，使得表单或工具栏部分或全部显露出时，重画该表单或工具栏。

在 Resize 事件中使用 Refresh 方法时，每次调整表单或工具栏大小时，都强制重画整个对象。

完成某种任务时使用 Paint 事件会引起级联事件。通常，在下列情况下应避免使用 Paint 事件：

- 移动表单或控件或调整它们的大小。
- 更改任何影响大小或外观的变量，例如设置一个对象的 BackColor 属性
- 调用 Refresh 方法。

注意 对于这些任务，调用 Resize 事件可能更合适。

应用于

表单，工具栏

请参阅

Circle **方法**, ClipControls **属性**, Line **方法**, Print **方法**, PSet **方法**, Refresh **方法**, Resize **事件**

Panel **属性**

指定一个表格控件中的活动窗格。设计时可用，运行时可读写。且仅当 Partition **属性的设置值大于** 0 **时该属性可用。**

语法

Grid.Panel[= nSide]

参数描述

nSide

Panel **属性的设置为：**

设置	说明
0	左窗格是活动的。
1	(默认值)右窗格是活动的。

应用于

表格

请参阅

PanelLink 属性, Partition 属性

PanelLink 属性

指定当拆分表格时，表格控件的左右窗格是否链接。设计时可用，运行时可读写。且仅当 Partition 属性的设置值大于 0 时该属性可用。

语法

Grid.PanelLink[= lExpr]

参数描述

lExpr

PanelLink 属性的设置为：

设置	说明
“真” (.T.)	(默认值)当拆分表格时左右窗格链接。
“假” (.F.)	当拆分表格时左右窗格不链接。

如果 PanelLink 设置为“真”(.T.)，当设置下列属性时，表格的两个窗格都受影响：HeaderHeight，DeleteMark，GridLineColor，GridLines，GridLineWidth，Highlight，RecordMark，RowHeight 和 ScrollBars。

如果 PanelLink 设置为“假”(.F.)，Panel 属性决定下列属性设置在哪个窗格上：HeaderHeight，DeleteMark，GridLineColor，GridLines，GridLineWidth，HighLight，RecordMark，RowHeight 和 ScrollBars。

应用于

表格

请参阅

DeleteMark 属性，GridLineColor 属性，GridLines 属性，GridLineWidth 属性，HeaderHeight 属性，Highlight 属性，Panel 属性，Partition 属性，RecordMark 属性，RowHeight 属性，ScrollBars 属性

PARAMETERS 命令

将调用程序传来的数据赋值给私有变量或数组。

语法

PARAMETERS ParameterList

参数描述

ParameterList

指定接收数据的变量或数组。

*“ParameterList”*中的参数应用逗号分隔。PARAMETERS语句中的参数数目至少应与DO ... WITH语句中的参数数目相同。如果PARAMETERS语句中所列的变量或数组数目比DO ... WITH语句传递的多，剩余的变量或数组就初始化为“假”(.F.)。最多可传递27个参数。

PARAMETERS() 函数返回传递给最近执行程序的参数数目。

说明

当PARAMETERS命令与DO ... WITH语句一起使用时，它必须是被调用程序、过程或用户自定义函数中的第一条可执行语句。

默认情况下，DO ... WITH语句以引用传递方式向过程传递变量和数组。当一个值在被调用过程中更改时，新值传递回调用程序中相关的变量或数组。如果想以值传递方式向一个过程传递变量或数组，在DO ... WITH语句的参数列表中用括号把变量或数组括起来，这样，在被调用过程中对参数的任何更改都不传回调用程序。

默认情况下，变量以引用传递方式向过程传递，以值传递方式向一个用户自定义函数传递。使用SET UDFPARMS TO REFERENCE命令，能以引用传递方式向用户自定义函数传递变量。

示例

下面的示例传递参数到错误处理例程中。

```
ON ERROR DO errhand WITH ERROR（）, MESSAGE（）,;
    MESSAGE(1),PROGRAM（）,LINENO（）
USE nodatabase
ON ERROR    && 恢复系统错误处理程序
PROCEDURE errhand
PARAMETERS gnError, gcMess, gnMess1, gcProg, gnLineNo
? 'Error number: ' + LTRIM(STR(gnError))
? 'Error message: ' + gcMess
? 'Line of code with error: ' + gnMess1
? 'Line number of error: '+ LTRIM(STR(gnLineNo))
? 'Program with error: ' + gcProg
```

请参阅

DO, LOCAL, LPARAMETERS, PARAMETERS（）, PRIVATE, PUBLIC, SETUDFPARMS

PARAMETERS（） 函数

返回传递给程序、过程或用户自定义函数的参数数目，这里是指最近调用的程序、过程或用户自定义函数。

语法

PARAMETERS（）

返值类型

数值型

说明

PARAMETERS（） 函数用以确定有多少参数传递给一个程序、过程，或用户自定义函数。

注意 每次调用程序、过程或用户自定义函数，或者执行 ON KEY LABEL 命令时，PARAMETERS（） 函数的返回值都要被重置。和 PARAMETERS（） 函数不同 PCOUNT（）函数不会被重置，所以大多数情况下使用 PCOUNT（）函数。

示例

示例 1 调用一个过程，并且在等待窗口中显示传递参数的数量。

示例 2 用一个过程来显示 4 个数值的平均数。

*示例 1

```
DO testpar WITH 1,2,3

PROCEDURE testpar
PARAMETERS gn1,gn2,gn3
gcMessage = 'PARAMETERS（） ='+ALLTRIM(STR(PARAMETERS（） ))
WAIT WINDOW (gcMessage)
RETURN
```

```
* 示例 2
SET TALK OFF
gnVal1 = 10
gnVal2 = 20
gnVal3 = 30
gnVal4 = 15
gnMin = getavg(gnVal1, gnVal2, gnVal3, gnVal4)
? 'Average value is '
?? gnMin

* 该用户定义函数允许通过 9 个参数
* 使用 PARAMETERS() 函数决定参数数量
* 全部通过并返回均值

FUNCTION getavg
PARAMETERS gnPara1,gnPara2,gnPara3,gnPara4,gnPara5, ;
    gnPara6,gnPara7,gnPara8,gnPara9
IF PARAMETERS() = 0
    RETURN 0
ENDIF
gnResult = 0
FOR gnCount = 1 to PARAMETERS()
    gcCompare = 'gnPara' +(STR(gnCount,1))
    gnResult = gnResult + EVAL(gcCompare)
ENDFOR
gnResult = gnResult / (gnCount - 1)
RETURN gnResult
```

请参阅

DO, FUNCTION, LPARAMETERS, LOCAL, PARAMETERS, PCOUNT（）, PRIVATE, PROCEDURE, PUBLIC, SET UDFPARMS

Parent Object 参考

引用一个控件的容器对象。设计时不可用，运行时只读。

语法

Control.Parent

说明

使用 Parent 属性，可以访问一个控件所属的容器对象的属性、方法或控件，还可以访问一个页面或表单的容器对象。

在应用程序中，当把控件作为参数传递时，Parent 属性很有用。例如，可以把一个控件变量传递给一个通用过程并且应用 Parent 属性访问此控件的容器对象。

如果一个控件类中的对象被放置在一个未知的表单对象上，也可以用 Parent 属性来引用。

应用于

ActiveDoc 对象，复选框，列，组合框，命令按钮，命令组，容器对象，控件对象，自定义，编辑框，表单，表格，标头，图像，标签，线条，列表框，OLE

绑定型控件，OLE 容器控件，选项按钮，选项组，页面，页框，项目对象，项目挂接对象，分隔符，形状，微调，文本框，计时器，工具栏

请参阅

BaseClass 属性

ParentAlias 属性

指定父表的别名。设计和运行时只读。

语法

DataEnvironment.Relation.ParentAlias[= cAliasName]

参数描述

cAliasName

指定关系中父表的别名。

说明

ParentAlias 属性的设置必须与代表父表的临时表对象的 Alias 属性设置相同。

应用于

关系

请参阅

Alias 属性, ChildAlias 属性

ParentClass 属性

返回对象所属类的基类。设计和运行时只读。

语法

Object.ParentClass

说明

在类设计器中， ParentClass **属性包含子类所属类的基类名称。**

应用于

ActiveDoc **对象，复选框，列，组合框，命令按钮，命令组，容器对象，控件对象，自定义，编辑框，表单，表单集，表格，标头，图像，标签，线条，列表框，** OLE **绑定型控件，** OLE **容器控件，选项按钮，选项组，页面，页框，项目挂接对象，** _SCREEN**，形状，微调，文本框，计时器，工具栏**

请参阅

Parent 属性

Partition 属性

指定一个表格是否拆分为两个窗格，并且指定相对于表格左边的拆分位置。设计时可用，运行时可读写。

语法

Grid.Partition[= nSplit]

参数描述

nSplit

指定表格控件中两个窗格的拆分位置，如果 *nSplit* 为 0，则不拆分表格。

说明

注意，SplitBar 属性的优先级比 Partition 属性高。

应用于

表格

请参阅

Panel 属性，PanelLink 属性，SplitBar 属性

PasswordChar 属性

决定用户输入的字符或占位符是否显示在文本框控件中，并确定用作占位符的字符。设计和运行时可用。

语法

TextBox.PasswordChar[= cCharString]

参数描述

cCharString

指定显示在文本框中的字符。

说明

使用这个属性，可以在对话框中创建一个口令字段。虽然可以使用任何字符，但 Windows **应用程序更经常地使用星号** (*)，**即** ANSI **码** 42。

这个属性不影响 Value **属性的设置；它忠实地包含用户键入的内容或者代码的设置值。为了显示实际文本，可把** PasswordChar **设置为空字符串** ("")。**默认设置值是空字符串。**

可以把这个属性指定为任何字符串，但只有第一个字符是有意义的，其他字符都被忽略。

应用于

文本框

请参阅

Format **属性**, InputMask **属性**, Value **属性**

PAYMENT（） **函数**

返回固定利息贷款按期兑付的每一笔支出数量。

语法

PAYMENT(nPrincipal, nInterestRate, nPayments)

返值类型

数值型

参数描述

nPrincipal

指定贷款的开始本金。

nInterestRate

指定每个阶段的固定利率。如果贷款是按月偿还的，而利率是按年计算的，

则把年利率除以 12。

nPayments

指定贷款的偿还次数。

说明

PAYMENT() 函数假定阶段利率是常数，并且假定在每个阶段偿还贷款。

示例

```
STORE 100000 to gnPrincipal  &&$100,000 的开始本金
STORE .105/12 TO gnInterest  &&10.5% 的年利率
STORE (20*12) TO gnPayments   && 20 年按月偿还
CLEAR
? PAYMENT(gnPrincipal, gnInterest, gnPayments) && 显示 998.38
```

请参阅

CALCULATE, FV(), PV()

_PBPAGE 系统变量

包含第一个要打印的页面。包含此变量是为了提供向后兼容性，可用报表设计器代替。

PCOL（） 函数

返回打印机打印头的当前列位置。

语法

PCOL（）

返值类型

数值型

说明

PCOL（） 函数的返回值与打印机左边距的当前设置值有关。可以使用 SET MARGIN 命令设置左边距，或把一个值存储于系统变量 _PLOFFSET 中。

PCOL（） 函数对欲打印文本的相对定位很有用。可用 $ 操作符代替 PCOL（） 函数。

示例

```
CLEAR
@ PROW（）, PCOL（） +12 SAY 'Contact person'
@ PROW（）, $+12 SAY 'Contact person'
```

请参阅

COL（），ROW（），PROW（）

_PCOLNO 系统变量

包含当前列的编号。包含此变量是为了提供向后兼容性。可用报表设计器代替。

_PCOPIES 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

PCOUNT（） 函数

返回传递给当前程序、过程或用户自定义函数的参数的个数。

语法

PCOUNT（）

返值类型

数值型

说明

PCOUNT（） 用于确定有多少参数传递给了当前程序、过程或用户自定义函数。

示例

示例 1 调用一个过程，并且在等待窗口中显示传递参数的数量。

示例 2 用一个过程来显示 4 个数值的平均数。

*示例 1

```
DO testpar WITH 1,2,3

PROCEDURE testpar
PARAMETERS gn1,gn2,gn3
gcMessage = 'PCOUNT（） ='+ALLTRIM(STR(PCOUNT（） ))
WAIT WINDOW (gcMessage)
RETURN
```

*示例 2

```
SET TALK OFF
gnVal1 = 10
gnVal2 = 20
gnVal3 = 30
gnVal4 = 15
```

```
gnMin = getavg(gnVal1, gnVal2, gnVal3, gnVal4)
? 'Average value is '
?? gnMin

*该自定义函数最多允许有 9 个参数
*使用 PCOUNT() 函数决定参数数量
*参数均通过后并返回平均值

FUNCTION getavg
PARAMETERS gnPara1,gnPara2,gnPara3,gnPara4,gnPara5, ;
    gnPara6,gnPara7,gnPara8,gnPara9
IF PCOUNT() = 0
    RETURN 0
ENDIF
gnResult = 0
FOR gnCount = 1 to PARAMETERS()
    gcCompare = 'gnPara' +(STR(gnCount,1))
    gnResult = gnResult + EVAL(gcCompare)
ENDFOR
gnResult = gnResult / (gnCount - 1)
RETURN gnResult
```

请参阅

DO, FUNCTION, LPARAMETERS, LOCAL, PARAMETERS, PARAMETERS(), PRIVATE, PROCEDURE, PUBLIC, SET UDFPARMS

_PDRIVER 系统变量

包含此变量是为了提供向后兼容性。应在 REPORT 中使用 TO FILE ASCII 参数。

_PDSETUP 系统变量

包含此变量是为了提供向后兼容性。可以在 REPORT 中使用 TO FILE ASCII 参数。

_PECODE 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

_PEJECT 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

PEMSTATUS( ) 函数

返回一个属性、事件、方法或对象的状态。

语法

PEMSTATUS(oObjectName | cClassName, cProperty | cEvent | cMethod | cObject,
nAttribute)

返值类型

字符型或逻辑值

参数描述

oObjectName
指定一个对象，返回其属性、事件、方法或对象的状态。*oObjectName* 可以是任意求值结果为对象的表达式，例如对象引用、对象变量或对象数组元素。如果 oObjectName 是一个容器对象，例如表单，就可以确定容器中对象的属性。

cClassName
指定一个类，返回其属性、事件或方法的状态。

cProperty
指定要返回其状态的属性。

cEvent
指定要返回其状态的事件。

cMethod
指定要返回其状态的方法。

cObject
指定要返回其状态的对象。例如，可以使用 AddObject 方法将一个对象添加到一个容器对象中，然后使用 PEMSTATUS（） 返回该对象的信息。

nAttribute
指定对应属性、事件或方法的状态的数值。

下表列出了与属性、事件或方法的状态对应的值。

nAttribute	属性、事件或方法的状态
0	已更改(仅用于属性)。如果属性的原始值、默认值已经更改，则返回“真”(.T.)，否则，返回“假”(.F.)。
1	只读(仅用于属性)。如果属性是只读的，则返回“真”(.T.)，否则，返回“假”(.F.)。
2	受保护的。如果属性、事件或方法是受保护的，则返回“真”(.T.)，否则，返回“假”(.F.)。
3	类型。返回一个字符串表示 *cProperty*、*cEvent*、*cMethod* 或 cObject 是否为属性、事件、方法或对象。返回 Property、Event、Method 或 Object。
4	用户自定义。如果属性、事件或方法是用户自定义的属性、事件或方法，则返回“真”(.T.)，否则，返回“假”(.F.)。
5	已定义的属性、事件、方法或对象。如果 *oObjectName* 或 *cClassName* 的属性、事件、方法或对象已存在，则返回“真”(.T.)，否则，返回“假”(.F.)。
6	继承的属性、事件、方法或对象。如果 oObjectName 或 cClassName 的属性、事件、方法或对象是从另一个对象或类继承的，就返回“真”(.T.)，否则返回“假”(.F.)。

说明

在一条 Visual FoxPro 命令中调用两次以上 PEMSTATUS（），会使一个表单变成失效。可以将该命令分成几个命令，每个命令包含一个 PEMSTATUS（）函数。

请参阅

CREATE FORM, GETPEM（）, SYS(1269), SYS(1270), SYS(1271), SYS(1272)

_PEPAGE 系统变量

包含向后兼容性。用“报表设计器”代替。

PI（） 函数

返回数值常数 PI。

语法

PI（）

返值类型

数值型

说明

数值型常量 pi (3.141592) **是圆的周长与直径的比率。**

PI（） **函数返回值的小数点位置由** SET DECIMALS **命令决定。**

示例

```
CLEAR
? PI（） && 显示 3.14
STORE 2.30 TO gnRadius
STORE PI（） * gnRadius^2 TO gnArea
? gnArea && 显示 16.6190
```

请参阅

SET DECIMALS

Picture 属性

指定在控件中显示的图形文件。

语法

Control.Picture[= cFileName]

参数描述

cFileName
指定一个 .bmp、.gif、.jpg 或 .ico 文件。

说明

此属性的设置值是一个 .bmp、.gif、.jpg 或 .ico 文件的路径。

注意 如果在设计时设置 Picture 属性但指定的文件不存在，Visual FoxPro 将显示错误信息，但仍把属性设置为指定的文件。如果运行时设置的文件不存在，Visual FoxPro 将忽略 Picture 属性。

图形在控制中居中对齐。对于复选框和选项按钮，Style 属性必须设置为 1（图形）。对于命令按钮，为了能在控制中显示位图，Style 属性必须设置为 0（标准）。

对于命令按钮和选项按钮控制，不管按钮可用、选中或不可用，都会使用指定位图。

可以使用 DownPicture 或 DisabledPicture 属性为每种按钮状态指定一个不同的位图。

应用于

复选框，命令按钮，容器对象，控件，自定义，表单，图像，选项按钮，页面，_SCREEN

请参阅

DisabledPicture 属性，DownPicture 属性，Style 属性

PLAY MACRO 命令

执行一个键盘宏。

语法

PLAY MACRO MacroName
[TIME nDelay]

参数描述

MacroName
指定需要执行的键盘宏名称。

TIME nDelay
指定在键盘宏中释放每个键击的时间间隔，延迟时间必须在 0 到 10 秒之间。*nDelay* 计算结果可以是十进制的小数。例如，如果指定 *nDelay* 等于 1.5，则将以 1.5 秒的时间间隔执行宏中的键击。

说明

通过按 SHIFT+F10，可以把一系列键击保存为一个键盘宏。PLAY MACRO 执行这一系列键击，在程序中执行键盘宏可以创建自动运行的演示程序。

如果在命令窗口中发出 PLAY MACRO 命令，将立即执行键盘宏。如果在程序中使用 PLAY MACRO 命令，直到程序执行一个允许键盘输入的命令之后才执行它。等待输入的命令有 @ ... GET，BROWSE，CHANGE 和 EDIT 等。

如果程序中有一系列的 PLAY MACRO 命令等待执行，Visual FoxPro 并不按它们出现的顺序执行。宏是按逆序执行的——第一个 PLAY MACRO 最后执行，最后一个 PLAY MACRO 最先执行——在程序中按照从底向上的顺序执行。

请参阅

CLEAR MACROS, ON KEY LABEL, RESTORE MACROS, SAVE MACROS

_PLENGTH 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

_PLINENO 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

_PLOFFSET 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

Point 方法

返回一个表单上特定点的红-绿-蓝 (RGB) 颜色。

语法

Object.Point ([nXCoord, nYCoord])

参数描述

nXCoord

按照表单 ScaleMode 属性设置的度量单位指定点的横坐标。

nYCoord

按照表单的 ScaleMode 属性设置的度量单位指定点的纵坐标。

说明

如果 *nXCoord* 或 *nYCoord* 在表单之外，则 Point 方法返回 -1。

应用于

表单，_SCREEN

请参阅

Circle 方法，Line 方法，Print 方法，PSet 方法

POP KEY 命令

恢复用 PUSH KEY 命令放入栈内的 ON KEY LABEL 指定键值。

语法

POP KEY

[ALL]

参数描述

ALL

清除所有当前用 ON KEY LABEL 命令定义的键指定值，并且清除堆栈中所有用 ON KEY LABEL 命令定义的键指定值。

说明

当 POP KEY 与 PUSH KEY 命令一起使用时，可以先保存一系列用 ON KEY LABEL 命令创建的键指定值，然后用更多的 ON KEY LABEL 命令更改键的指定值，最后还原初始的键指定值。

有关在堆栈上使用 ON KEY LABEL 命令的详细内容，请参阅 PUSH KEY。

请参阅

ON KEY LABEL, ON（）, PUSH KEY

POP MENU 命令

恢复指定的菜单栏定义，该定义曾被 PUSHMENU 放在堆栈中。

语法

POP MENU MenuBarName
[TO MASTER]

参数描述

MenuBarName

指定菜单栏的名称，该菜单栏的定义被放在堆栈中。指定的菜单栏可以是用户自定义的也可以是 Visual FoxPro 系统菜单栏。

TO MASTER

恢复放在堆栈中的第一个菜单栏定义，然后清空堆栈。

说明

当与 PUSH MENU 一起使用时，POP MENU 允许保存一个菜单栏定义，更改这个菜单栏定义，然后将菜单栏定义恢复到原来状态。

菜单栏按照后进先出顺序放进和移出堆栈。

示例

下面的示例将系统菜单栏定义放进堆栈，然后修改它。接着通过出栈来恢复原始的系统菜单定义。

```
PUSH MENU _MSYSMENU
SET SYSMENU TO _MFILE, _MEDIT
POP MENU _MSYSMENU
```

请参阅

ACTIVATE MENU, DEFINE MENU, POP POPUP, PUSH MENU, PUSH

POP POPUP 命令

恢复用 PUSH POPUP 命令压进堆栈的菜单定义。

语法

POP POPUP MenuName

参数描述

MenuName

指定菜单名，从堆栈中弹出该菜单定义。这个菜单可以是用 DEFINE MENU 命令创建的用户自定义菜单，或者是 Visual FoxPro 系统菜单。

说明

POP POPUP 与 PUSH POPUP 一起使用时，可以先保存一个菜单定义，然后修改这个菜单定义，最后把它恢复到初始状态。

菜单定义按照后进先出顺序被压进和弹出堆栈。

示例

下面的示例创建了菜单 popExam。菜单定义进入堆栈，然后修改一个菜单项。接着通

过出栈来恢复原始的菜单定义。

```
DEFINE POPUP popExam FROM 5,5
DEFINE BAR 1 OF popExam PROMPT 'One'
DEFINE BAR 2 OF popExam PROMPT 'Two'
DEFINE BAR 3 OF popExam PROMPT 'Three'
DEFINE BAR 4 OF popExam PROMPT 'Four'
ACTIVATE POPUP popExam NOWAIT

PUSH POPUP popExam
WAIT 'Popup pushed' WINDOW
RELEASE BAR 2 OF popExam
WAIT 'This is the modified popup' WINDOW

POP POPUP popExam
WAIT 'Popup popped, original popup restored' WINDOW
DEACTIVATE POPUP popExam
RELEASE POPUP popExam
```

请参阅

ACTIVATE POPUP, DEFINE POPUP, POP MENU, PUSH MENU, PUSH POPUP

POPUP（） 函数

以字符串形式返回活动菜单名。或者返回一个逻辑值，该值表明是否定义了一个菜单。

语法

POPUP([cMenuName])

返值类型

字符型或逻辑型

参数描述

cMenuName

返回一个逻辑值，表明 *cMenuname* 是否已经定义。如果指定的菜单已经定义，POPUP（） 函数返回“真”(.T.)；否则 POPUP（） 函数返回“假”(.F.)。

说明

如果省略可选参数 *cMenuName*，POPUP（） 函数以字符串形式返回活动菜单名。要使 POPUP（） 函数返回菜单名，必须定义且激活一个菜单。可用 DEFINE POPUP 和 ACTIVATE POPUP 命令创建和激活菜单。菜单也可以是 Visual FoxPro 系统菜单。如

果没有定义菜单或菜单不活动，或者在命令窗口发出 POPUP（） 函数，POPUP（）将返回一个空字符串。

请参阅

ACTIVATE POPUP, BAR（）, DEFINE BAR, DEFINE POPUP, ON SELECTION POPUP

_PPITCH 系统变量

包含此变量是为了提供向后兼容性，可用报表设计器代替。

_PQUALITY 系统变量

包含此变量是为了提供向后兼容性，可用报表设计器代替。

_PRETEXT 系统变量

指定一个放在文本合并行开头的字符表达式。

语法

_PRETEXT = cExpression

说明

用 SET TEXTMERGE 命令可以在一个程序中合并文本。可以向屏幕、窗口、打印机、文本文件或低级文件输出文本行、变量的内容以及表达式和函数的结果。

如果把一个字符表达式存储在 _PRETEXT 中，该字符表达式就会加在输出文本行的开头。为了方便程序缩进，可以把一个或多个制表符存储在 _PRETEXT 中。

请参阅

\ | \\, SET TEXTMERGE, _TEXT

PRIMARY（） 函数

检查索引标识，如果为主索引标识，就返回“真”(.T.)；否则返回“假”(.F.)。

语法

PRIMARY([nIndexNumber] [, nWorkArea | cTableAlias])

返值类型

逻辑值

参数描述

nIndexNumber

指定索引标识的编号，PRIMARY（） 函数返回该索引是否为主索引。

指定 *nIndexNumber* 是从 1 到结构复合索引标识和独立复合索引标识的总数间的某个数时，PRIMARY（） 函数按下列顺序返回主索引状态：

1. 首先返回结构复合索引（如果存在）中每个标识的主索引状态，主索引状态按照标识在结构复合索引中创建的顺序返回。
2. 然后返回独立复合索引中每个标识的主索引状态，该独立复合索引文件必须是打开的。主索引状态按照标识在独立复合索引中创建的顺序返回。

如果省略 *nIndexNumber* 参数，PRIMARY（） 将检查主控索引标识是否为主索引标

识。如果没有主控索引标识，PRIMARY（） 将返回“假”(.F.)。

nWorkArea

指定索引标识的工作区。

cTableAlias

指定索引标识的表别名。

如果省略 *nWorkArea* 和 *cTableAlias* 参数。PRIMARY（） 检查最近选定的工作区中的索引标识，看它是否为主索引标识。

示例

下面的 example 打开 testdata 数据库中的 customer 表，应用 FOR ... ENDFOR 语句创建一个循环，在此循环中检查 customer 结构索引中每个索引标识的主索引状态，然后显示每个结构索引标识名称以及其主索引状态。

```
CLOSE DATABASES
SET PATH TO (HOME(2) + 'Data\')&& 设置数据库路径
OPEN DATABASE testdata  && 打开 testdata 数据库
USE Customer  && 打开 customer 表

FOR nCount = 1 TO 254
   IF !EMPTY(TAG(nCount))  && 检查索引标识
   ? TAG(nCount)  && 显示标识名
   ? PRIMARY(nCount)  && 显示主索引状态
   ELSE
      EXIT  && 当没发现更多标识时退出循环
   ENDIF
ENDFOR
```

请参阅

ALTER TABLE – SQL, CANDIDATE( ), CREATE TABLE – SQL, INDEX

Print 方法

在表单对象上打印一个字符串。

语法

[FormSet.] Object.Print [(cText)]

参数描述

cText

指定需要打印的字符。如果省略 *cText* **参数，就打印一个空行。**

在表单上打印的初始位置由 CurentX **和** CurrentY **属性指定。如果包含一个回车符，**CurrentX **按** *cText* **的宽度值（与** TextWidth **的返回值相同）增加，而** CurrentY **保持不变。**

当 Print **方法结束时，**CurrentX **和** CurrentY **属性设置成紧跟在最终打印字符后面的点。**

应用于

表单，_SCREEN

请参阅

CurrentX, CurrentY 属性, TextHeight 方法, TextWidth 方法

PRINTJOB ... ENDPRINTJOB 命令

激活打印作业中系统变量的设置。

语法

PRINTJOB

Commands

ENDPRINTJOB

参数描述

Commands

在打印作业结束之前，要执行的 Visual FoxPro 命令。

说明

PRINTJOB ... ENDPRINTJOB 初始化打印机以及一些影响打印输出的系统变量。它可以向打印机传送控制代码、在打印作业之前和（或）之后走纸、初始化打印机列数、

控制打印的数目。

PRINTJOB 执行下列任务：

- 向打印机传送启动控制代码，它们存储在系统变量 _PSCODE 中。有关打印机控制代码的详细内容，请参阅稍后的“系统变量概述”和您的打印机手册。
- 如果系统变量 _PEJECT 设置为 BEFORE 或 BOTH，则走一页纸。

把系统变量 _PCOLNO 设置为 0。_PCOLNO 存储打印机列数。

ENDPRINTJOB 执行下列任务：

- 向打印机传送打印机结束控制代码，它们存储在系统变量 _PECODE 中。可以把打印机重新设置为执行 PRINTJOB 前的配置。
- 如果系统变量 _PEJECT 设置为 AFTER 或 BOTH，则走一页纸。
- 如果系统变量 _PCOPIES 设置为大于 1（默认值）的值，则循环到 PRINTJOB 语句来开始打印报表的另一个副本，系统变量 _PCOPIES 的值决定了副本数目。当已打印的副本数目等于 _PCOPIES 值时，Visual FoxPro 退出循环，程序执行 ENDPRINTJOB 后的下一条命令。

PRINTJOB 和 ENDPRINTJOB 只能在一个程序内部执行，不能嵌套 PRINTJOB…… ENDPRINTJOB 命令。

请参阅

ON PAGE, **系统变量概述**

PRINTSTATUS（） 函数

如果打印机或打印设备已联机，则返回“真”(.T.)；否则返回“假”(.F.)。

语法

PRINTSTATUS（）

返值类型

逻辑值

说明

PRINTSTATUS（） 与 SYS(13) 相类似，不同的是 SYS(13) 返回 READY 而不是“真”(.T.)，返回 OFFLINE 而不是“假”(.F.)。

在 Visual FoxPro 中，如果已通过 Windows 的“控制面板”将打印机联机，则 PRINTSTATUS（） 总返回“真”(.T.)。

示例

```
? PRINTSTATUS（）
*** 程序 example***
STORE PRINTSTATUS（） TO glReady
IF NOT glReady
   WAIT '请确认已连接并打开了打印机！' WINDOW
ELSE
   WAIT '打印机已准备就绪！' WINDOW
```

ENDIF

请参阅

SET DEVICE, SET PRINTER, SYS(13)

PRIVATE 命令

在当前程序中隐藏指定的、在调用程序中定义的变量或数组。

语法

PRIVATE VarList

–或者–

PRIVATE ALL

[LIKE Skeleton | EXCEPT Skeleton]

参数描述

VarList

要声明为私有的变量或数组。

ALL LIKE Skeleton

PRIVATE **隐藏所有名称与** *Skeleton* **相匹配的变量和数组，** *Skeleton* **可以包含**

问号 (?) 和星号 (*) 通配符。

ALL EXCEPT Skeleton

PRIVATE 隐藏所有名称与 *Skeleton* 不匹配的变量或数组，*Skeleton* 可以包含问号 (?) 和星号 (*) 通配符。

说明

将高层程序中创建的、与私有变量同名的变量隐藏起来，可以在当前程序中操作这些私有变量，而不影响被隐藏变量的值。一旦包含 PRIVATE 命令的程序执行完毕，所有声明为私有的变量和数组就可恢复使用。

PRIVATE 并不创建变量，它只在当前程序中隐藏变量，这些变量是在高层程序中声明的。

示例

```
*** 程序示例演示 PRIVATE ***
SET TALK OFF
val1 = 10
val2 = 15

DO down
? val1, val2  && 显示 10, 100

PROCEDURE down
PRIVATE val1
val1 = 50
val2 = 100
? '  Val1    Val2'
? val1, val2  && 显示 50, 100
```

RETURN

请参阅

DIMENSION, FUNCTION, LOCAL, LPARAMETERS, PARAMETERS, PARAMETERS（）, PROCEDURE, PUBLIC

PRMBAR（） 函数

返回一个菜单项的文本。

语法

PRMBAR(MenuName, nMenuItemNumber)

返值类型

字符型

参数描述

MenuName

指定菜单名。

nMenuItemNumber

菜单项的编号， PRMBAR（） **返回它的文本。例如，如果**

nMenuItemNumber 为 1，就返回第一个菜单项的文本；如果 *nMenuItemNumber* 为 2，就返回第二个菜单项的文本，… 依此类推。这个表达式至少为 1，并且不能大于菜单中菜单项的数目。

说明

可以用 DEFINE POPUP 命令创建菜单，用 DEFINE BAR 命令创建菜单中的菜单项。PRMBAR（） 也可以用于 Visual FoxPro 系统菜单。PRMBAR（） 函数返回出现在菜单项上的文本，菜单可以不是活动的。

如果创建菜单时，用反斜杠和小于号 (\<) 创建了访问键，或用反斜杠 (\) 使菜单项废止，PRMBAR（） 只返回菜单项的文本，而不包含这些特殊字符。当菜单项是用反斜杠和破折号 (\-) 创建的分隔线时，PRMBAR（） 返回一个空字符串。

请参阅

CNTBAR（）, GETBAR（）, DEFINE BAR, DEFINE POPUP, MRKBAR（）, PRMPAD（）

PRMPAD（） 函数

返回一个菜单标题的文本。

语法

PRMPAD(MenuBarName, MenuTitleName)

返值类型

字符型

参数描述

MenuBarName

指定包含菜单标题的菜单栏名称。

MenuTitleName

指定菜单标题。

说明

可用 DEFINE MENU 命令创建菜单栏，用 DEFINE PAD 命令创建菜单标题。

PRMPAD（） 函数也可以用于 Visual FoxPro 系统菜单。PRMPAD（） 返回一个菜单标题的文本，该菜单栏可以不是活动的。

如果创建菜单标题时用反斜杠和小于号 (\<) 创建了访问键，或用反斜杠 (\) 使菜单标题废止，PRMPAD（） 只返回菜单标题的文本，而不包含这些特殊字符。

示例

下面的示例创建了有三个菜单标题的菜单栏 mnuexample 。从菜单标题 titleTwo 与 titleThree 中不返回访问关键字和禁用的选项指示符。激活菜单栏以显示菜单标题，并且在选择了菜单标题后，从屏幕和内存中清除它。

```
CLEAR
```

```
SET TALK OFF
STORE 'mnuexample' TO gcPopName

DEFINE MENU mnuexample BAR AT LINE 1
DEFINE PAD titleOne OF mnuexample PROMPT 'This will be returned'
DEFINE PAD titleTwo OF mnuexample PROMPT '\<As will this'
DEFINE PAD titleThree OF mnuexample PROMPT '\And this, too'

=messagebox( PRMPAD('mnuexample', 'titleOne') )
=messagebox( PRMPAD('mnuexample', 'titleTwo') )
=messagebox( PRMPAD(gcPopName, 'titleThree') )

ACTIVATE MENU mnuexample
DEACTIVATE MENU mnuexample
RELEASE MENU mnuexample
```

请参阅

CNTBAR（），GETBAR（），DEFINE BAR, DEFINE POPUP, MRKPAD（），PRMBAR（）

PROCEDURE 命令

在程序文件中，标识一个过程的开始。

语法

```
PROCEDURE ProcedureName
  Commands
  [RETURN [eExpression ]]
ENDPROC
```

参数描述

ProcedureName

指定要创建的过程名称。

说明

PROCEDURE 是程序文件中的语句，它指明程序文件中每个过程的开始，并且定义过程名称。过程名称必须以一个字母或下划线开头，可以包含字母、数字和下划线的任何组合。

在 Visual FoxPro 中，过程名最长可达 254 个字符。

注释可放在同一行的 PROCEDURE 和 ENDPROC 后。在编译和执行程序时，将忽略注释。

PROCEDURE 语句后有一系列的命令，以构成过程。在过程的任何地方都可以包含 RETURN 以返回控制到调用程序或另一个程序，还可以定义过程返回的值。如果没有包含 RETURN 命令，在函数退出时会自动执行一个隐含的 RETURN 命令。如果 RETURN 命令没有包含返回值（或执行一个隐含的 RETURN）时，Visual FoxPro 将 .T.（真）作为返回值。

过程以 ENDPROC 命令结尾。此命令是可选的；过程遇到 PROCEDURE 命令、FUNCTION 命令或程序文件的结尾时退出。

注意 您不能在过程后的程序文件中包含普通的执行程序代码，在文件中的第一个 PROCEDURE 或 FUNCTION 命令后只能跟随过程、用户自定义函数和类定义。

当用 DO *ProcedureName* 命令执行过程时，Visual FoxPro 按特定的顺序搜索过程。这个搜索顺序是：

1. 包含 DO *ProcedureName* 命令的程序。
1. 当前的数据库。
2. 用 SET PROCEDURE 命令打开的过程文件。
3. 执行链中的程序。Visual FoxPro 从最近执行的程序开始，往回连续搜索到第一个执行的程序。
4. 一个独立的程序文件。如果找到了一个与 DO 所指定的文件名同名的程序文件，Visual FoxPro 就执行这个程序；如果没有找到匹配的程序文件名，Visual FoxPro 就产生错误信息。

在 DO 中包含 IN 子句以执行指定文件中的过程。

默认情况下，参数是以值传递到过程的。有关引用传递参数到过程的内容，请参阅稍后的 SET UDFPARMS 命令。最多可传递 27 个参数到过程中。在过程中包含 PARAMETERS 或 LPARAMETERS 语句，或者在 PROCEDURE *ProcedureName* 后随即放置一列参数，这样就可以传递参数到过程中。将参数放在一对括号中，并用逗号分开。

示例

下面的示例说明了如何调用过程来完成不连续的任务，如在日志文件中做入口。此过程打开日志文件（假设在本示例中存在），在基于参数传递的信息中构造入口，写出入口，并且关闭文件。用类似于程序最顶部的 DO 命令来调用过程。

```
DO MakeLogEntry WITH "Logged in", "jsmith"

PROCEDURE MakeLogEntry
    PARAMETERS message, username
    pnHandle = FOPEN("LOG2.TXT",2)      && 假定文件存在
    pnSize = FSEEK(pnHandle,0,2)        && 移到文件末
    logEntry = dtoc(date())+","+time()+","+username+","+message
    =FPUTS(pnHandle, logEntry)
    =FCLOSE(pnHandle)  && 关闭文件
ENDPROC
```

下面的示例显示了如何调用过程以返回值。

```
SET CENTURY ON
? longdate(({^1998-02-16})) && 显示周一， 1998 年 2 月 16 日

PROCEDURE longdate
    PARAMETER mdate
    RETURN CDOW(mdate) + ", " + MDY(mdate)
ENDPROC
```

请参阅

FUNCTION, LPARAMETERS, LOCAL, PARAMETERS, PARAMETERS(), PRIVATE, PUBLIC, RETURN, SET PROCEDURE, SET UDFPARMS

返回总目录

PUSH MENU 命令
PUSH POPUP 命令
PUTFILE（）函数
PV（）函数
_PWAIT 系统变量
QueryAddFile 事件
QueryModifyFile 事件
QueryRemoveFile 事件
QueryRunFile 事件
QueryUnload 事件
QUIT 命令
Quit 方法
RAND（）函数
RangeHigh 事件
RangeLow 事件
RAT（）函数
RATC（）函数
RATLINE（）函数
RD | RMDIR 命令
RDLEVEL（）函数
READ EVENTS 命令

READ 命令
READ MENU 命令
ReadActivate 事件
ReadBackColor, ReadForeColor 属性
ReadCycle 属性
ReadDeactivate 事件
ReadExpression 方法
READKEY（）函数
ReadLock 属性
ReadMethod 方法
ReadMouse 属性
ReadObject 属性
ReadOnly 属性
ReadSave 属性
ReadShow 事件
ReadTimeout 属性
ReadValid 事件
ReadWhen 事件
RECALL 命令
RECCOUNT（）函数
RECNO（）函数

RecordMark 属 性
RecordSource 属 性
RecordSourceType 属 性
RECSIZE（） 函 数
Refresh 方 法
REFRESH（） 函 数
REGIONAL 命 令
REINDEX 命 令
Relation 对 象
RELATION（） 函 数
RelationalExpr 属 性
RelativeColumn 属 性
RelativeRow 属 性
RELEASE 命 令
RELEASE BAR 命 令
RELEASE CLASSLIB 命 令
RELEASE LIBRARY 命 令
RELEASE MENUS 命 令
Release 方 法
RELEASE PAD 命 令
RELEASE POPUPS 命 令

RELEASE PROCEDURE 命令
RELEASE WINDOWS 命令
ReleaseType 属性

ProgID 属性

包含项目中一个服务程序的注册 PROGID (Programmatic Identifier)。设计和运行时只读。

语法

Object.ProgID

说明

PROGID 是在根据项目连编一个可执行文件 (.exe) 或动态链接库 (.dll) 时，在 Windows 注册表中为一个服务程序创建的。

应用于

服务程序对象

请参阅

CLSID 属性, CREATEOBJECTEX(), TypeLibCLSID 属性, TypeLibDesc 属性, TypeLibName 属性

PROGRAM（）函数

返回当前正在执行的程序的名称、当前的程序级别，或者错误发生时执行程序的名称。

语法

PROGRAM([nLevel])

返值类型

字符型或者数值型

参数描述

nLevel

指定程序嵌套层次，这个参数值在 0 到程序嵌套深度之间。一个程序可以执行另一个程序，这个程序又可以执行另外一个程序，依次类推，最多可以嵌套 128 层。

如果把 *nLevel* 指定为 0 或 1，PROGRAM（）函数返回主程序名（最高层程序)。如果 *nLevel* 超过程序嵌套深度，PROGRAM（）返回一个空字符串。

如果把 *nLevel* 指定为 -1，PROGRAM（）函数返回一个数值表明当前程序级别。当从命令窗口中使用 PROGRAM(–1) 时，总是返回零。

说明

PROGRAM（）有助于程序从错误中恢复，它类似于 SYS(16)。

如果不使用参数，PROGRAM（）返回当前执行的程序名。

示例

```
ON ERROR DO errhand WITH PROGRAM( )
*** 下面一行应该生成一个错误***
USE nodatabase
ON ERROR     && 返回默认的系统错误处理程序
PROCEDURE errhand
PARAMETERS gcProgram
WAIT 'An error occurred in the program ' + gcProgram WINDOW
```

请参阅

DO, INENO( ), MESSAGE( ), SYS(16)

ProgrammaticChange 事件

在代码中更改一个控件值时发生。

语法

PROCEDURE Control.ProgrammaticChange

[LPARAMETERS nIndex]

参数描述

nIndex

在控件数组中唯一标识一个控件。

应用于

复选框，组合框，命令按钮，编辑框，列表框，选项组，微调，文本框

请参阅

Caption 属性, Style 属性

Projects 集合

项目对象的一个集合。

语法

Projects

说明

通过 Projects 集合来访问项目对象，允许您管理项目，以及项目中的文件和服务程序。

Projects 集合中的项可通过索引号或名称引用。例如，下面的代码重建最近打开的项目：

```
_VFP.Projects(1).Build( )
```

下面的代码重建项目 MyProject:

```
_VFP.Projects('MyProject.pjx').Build( )
```

有关 Projects 集合的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

属性

Count

方法

Item

请参阅

File 对象, Files 集合, Project 对象, ProjectHook 对象, Server 对象, Servers 集合

Project 对象

当创建或打开项目时实例化。

语法

Project

说明

当发出 CREATE PROJECT、MODIFY PROJECT、BUILD APP、BUILD DLL、BUILD EXE 或 BUILD PROJECT 命令时，实例化项目对象。

注意，在项目对象的 Init 事件中，项目对象和它的属性和方法不可用。会忽略在 Init 事件中设置的项目对象属性值或激活的方法。

注意，项目对象是一个 COM 对象，将项目对象的引用分配给一个变量，会创建“未知类型”类的变量。

有关项目的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

属性

Application	AutoIncrement	BaseClass
BuildDateTime	Debug	Encrypted
HomeDir	Icon	MainClass

续 表

MainFile	Name	Parent
ProjectHook	ProjectHookClass	ProjectHookLibrary
SCCProvider	ServerHelpFile	ServerProject
TypeLibCLSID	TypeLibDesc	TypeLibName
VersionComments	VersionCompany	VersionCopyright
VersionDescription	VersionLanguage	VersionNumber
VersionProduct	VersionTrademarks	Visible

方 法

Build	CleanUp	Close
Refresh	SetMain	

请 参 阅

ActiveProject **属 性**, File **对 象**, Files **集 合**, Projects **集 合**, ProjectHook **对 象**, Server **对 象**,
Servers **集 合**

ProjectHook 对象

每当打开一个项目时实例化，提供了对项目事件的编程访问。

语法

ProjectHook

说明

ProjectHook 对象是一个 Visual FoxPro 基类，在默认情况下，每当打开一个项目时实例化（在 CREATE PROJECT 和 MODIFY PROJECT 中包含 NOPROJECTHOOK 子句可以防止为项目实例化一个 ProjectHook 对象）。

ProjectHook 对象允许通过编程访问发生在项目中的事件。例如，每当将一个文件添加到项目中时都执行代码。注意，在“选项”对话框的“项目”选项卡中，可以为一个新项目指定默认的项目挂接类，或者在“项目信息”对话框中，可以为单个的项目指定默认的项目挂接类。在运行时，可以使用 ProjectHook 属性为一个项目指定项目挂接类。

```
MODIFY PROJECT MyProject
_VFP.Projects('MyProject.pjx').ProjectHook = ;
    NewObject('MyProjectHook', 'MyClass.vcx')
```

可以使用 CREATE CLASS 或 CREATEOBJECT() 创建 ProjectHook 基类。

有关项目的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

属性

Application	BaseClass	Class
ClassLibrary	Comment	Name
OLEDropEffects	OLEDropHasData	OLEDropMode
Parent	ParentClass	Tag

事件

AfterBuild	BeforeBuild	Destroy
Error	Init	OLEDragDrop
OLEDragOver	OLEGiveFeedBack	QueryAddFile
QueryModifyFile	QueryRemoveFile	QueryRunFile

方法

AddProperty	ReadExpression	ReadMethod
ResetToDefault	SaveAsClass	WriteExpression

请参阅

CREATE CLASS, CREATEOBJECT( ), File **对象**, Files **集合**, NEWOBJECT( ), ProjectHook **属性**, Projects **集合**, Project **对象**, Server **对象**, Servers **集合**

ProjectHook 属性

对实例化的 ProjectHook 对象的一个对象引用。

语法

Object.ProjectHook[= oProjectHookClass]

参数描述

oProjectHookClass

指定一个基于 Visual FoxPro ProjectHook 基类的类。在默认情况下，包含对默认的 ProjectHook 类的一个对象引用，该类是在“项目信息”对话框的“项目”选项卡中指定的。

说明

如果该属性没有分配一个基于 ProjectHook 基类的类，或者项目没有默认的 ProjectHook 类（是在“项目信息”对话框的“项目”选项卡中指定的），则 ProjectHook 属性包含 null 值。 将 null 值指定给 ProjectHook 属性，会释放这个 ProjectHook 对象，但是不影响项目的默认 ProjectHook 类。

有关项目的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

应用于

Project **对象**

请参阅

ProjectHook **对象**, ProjectHookClass **属性**, ProjectHookLibrary **属性**

ProjectHookClass **属性**

项目的默认 ProjectHook **类。**

语法

Object.ProjectHookClass[= cClassName]

参数描述

cClassName

指定项目的默认 ProjectHook **类。在为项目指定** ProjectHook **类之前，需要使用** ProjectHookLibrary **属性指定** .vcx **可视类库，其中包含** ProjectHook **类。**

若要删除项目的默认 ProjectHook **类，可将** ProjectHookClass **或** ProjectHookLibrary **属性设置为空字符串。**

说明

在“项目信息”对话框的“项目”选项卡中也可以为项目指定默认 ProjectHook 类。更改 ProjectHookClass 属性不能实例化新的 ProjectHook 对象。下一次打开项目时更改才生效。使用 ProjectHook 属性可以更改当前的 ProjectHook 对象。

有关项目的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

应用于

项目对象

请参阅

ProjectHook 对象, ProjectHook 属性, ProjectHookLibrary 属性

ProjectHookLibrary 属性

包含 .vcx 可视类库，在该类库中包含项目的默认 ProjectHook 类。

语法

Object.ProjectHookLibrary[= cLibraryName]

参数描述

cLibraryName

指定一个 .vcx 可视类库，其中包含一个基于 ProjectHook 基类的类。在使用这个属性指定了 .vcx 可视类库后，可使用 ProjectHookClass 属性指定项目的默认类。

若要删除项目的默认 ProjectHook 类，可将 ProjectHookClass 或 ProjectHookLibrary 属性设置为空字符串。

说明

在“项目信息”对话框的“项目”选项卡中也可以为项目指定默认 ProjectHook 类。

更改 ProjectHookLibrary 属性不能实例化新的 ProjectHook 对象。下一次打开项目时更改才生效。使用 ProjectHook 属性可以更改当前的 ProjectHook 对象。

有关项目的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第三十二章“应用程序开发和开发者的生产率”中的“项目管理器挂接程序”。

应用于

Project 对象

请参阅

ProjectHook 对象，ProjectHook 属性，ProjectHookClass 属性

PROMPT（）函数

返回菜单栏中选定的菜单标题，或者返回选定菜单项的文本。

语法

PROMPT( )

返值类型

字符型

说明

PROMPT（）函数返回菜单栏中最近选定的菜单标题文本，或者是一个菜单中最近选定的菜单项文本。菜单栏或菜单可以是用户自定义的菜单栏和菜单，也可以是 Visual FoxPro 的系统菜单栏或菜单。如果菜单栏或菜单不是活动的，或者按了 ESC 键退出菜单栏或菜单，PROMPT（）将返回一个空字符串。

可以用 DEFINE MENU 命令创建菜单栏，用 DEFINE PAD 命令创建菜单栏中的每个菜单标题。用 DEFINE POPUP 命令创建菜单，用 DEFINE BAR 命令创建菜单中的菜单项。

当菜单栏（或菜单）中的菜单标题（或菜单项）被选中时，MENU（）和 POPUP（）函数可以返回菜单栏名（或菜单名）。

请参阅

BAR( ), DEFINE BAR, DEFINE MENU, DEFINE PAD, DEFINE POPUP, POPUP( ), PRMBAR( )

PROPER（）函数

从字符表达式中返回一个字符串，字符串的每个首字母大写。

语法

PROPER(cExpression)

返值类型

字符型

参数描述

cExpression

指定的字符表达式，PROPER（）**函数返回首字母大写的字符串。**

示例

```
STORE 'Visual FoxPro ' TO gcExpr1
CLEAR
? PROPER(gcExpr1) && 显示 "Visual Foxpro "
```

```
STORE 'VISUAL FOXPRO ' TO gcExpr2
? PROPER(gcExpr2) && 显示 "Visual Foxpro "
```

请参阅

LOWER( ), UPPER( )

PROW（）函数

返回打印机中打印头的当前行号。

语法

PROW( )

返值类型

数值型

说明

如果执行 EJECT **命令，** Visual FoxPro **把** PROW（）**重置为** 0。

PROW（）**对于确定打印文本的相对位置特别有用。**

示例

在下面的 example **中，两个命令返回相同的结果。可以用** $ **操作符代替** PCOL（）**函**

数，$ 和 PCOL（）都返回打印机的当前列位置。

```
@ PROW( ), PCOL（） + 12 SAY ‘Contact person’
@ PROW( ), $+12 SAY ‘Contact person’
```

请参阅

@ & , COL( ), PCOL( ), ROW( )

PRTINFO（）函数

返回当前的打印机设置。

语法

PRTINFO(nPrinter 设置 [, cPrinterName])

返值类型

数值型

参数描述

nPrinter 设置

指定返回 Visual FoxPro 打印机的某个设置。

在下面表中列出了打印机设置的返回值。

注意 如果 PRTINFO(2) 返回 -1，就用 PRTINFO(3) 和 PRTINFO(4) 返回纸的大小。

如果 *nPrinterSetting* 等于 1 (FOXPRO.H 中的 PRT_ORIENTATION)，PRTINFO（）返回纸的方向。

返回值	设置
–1	信息不可使用
0	纵向
1	横向

如果 *nPrinterSetting* 等于 2 (FOXPRO.H 中的 PRT_PAPERSIZE)，PRTINFO（）返回纸的大小：

返回值	设置
–1 或除下列值	信息不可使用。使用 *nPrinterSetting* =3 和 *nPrinterSetting* =4 决定纸的大小。
1	Letter, 8 1/2 x 11 in
2	Letter 小, 8 1/2 x 11 in
3	Tabloid, 11 x 17 in
4	Ledger, 17 x 11 in
5	Legal, 8 1/2 x 14 in
6	Statement, 5 1/2 x 8 1/2 in
7	Executive, 7 1/4 x 10 1/2 in

续表

8	A3, 297 x 420 mm
9	A4, 210 x 297 mm
10	A4, 小 210 x 297 mm
11	A5, 148 x 210 mm
12	B4, 250 x 354 mm
13	B5, 182 x 257 mm
14	Folio, 8 1/2 x 13 in
15	Quarto, 215 x 275 mm
16	10 x 14 in
17	11 x 17 in
18	Note, 8 1/2 x 11 in
19	信封 #9, 3 7/8 x 8 7/8 in
20	信封 #10, 4 1/8 x 9 1/2 in
21	信封#11, 4 1/2 x 10 3/8 in
22	信封 #12, 4 1/2 x 11 in
23	信封 #14, 5 x 11 1/2 in
24	C 大小纸
25	D 大小纸
26	E 大小纸
27	信封 DL, 110 x 220 mm

续表

28	信封 C5, 162 x 229 mm
29	信封 C3, 324 x 458 mm
30	信封 C4, 229 x 324 mm
31	信封 C6, 114 x 162 mm
32	信封 C65, 114 x 229 mm
33	信封 B4, 250 x 353 mm
34	信封 B5, 176 x 250 mm
35	信封 B6, 176 x 125 mm
36	信封， 110 x 230 mm
37	信封 Monarch, 3 7/8 x 7.5 in
38	6 3/4 Envelope, 3 5/8 x 6 1/2 in
39	US Std Fanfold, 14 7/8 x 11 in
40	German Std Fanfold, 8 1/2 x 12 in
41	German Legal Fanfold, 8 1/2 x 13 in

如果 *nPrinterSetting* 等于 3 (FOXPRO.H 中的 PRT_PAPERLENGTH)，PRTINFO（）按 1mm 单位返回纸的长度。

如果 *nPrinterSetting* 等于 4 (FOXPRO.H 中的 PRT_PAPERWIDTH)，PRTINFO（）按 1mm 单位返回纸的宽度。

如果 *nPrinterSetting* 等于 5 (FOXPRO.H 的 PRT_SCALE)，PRTINFO（）返回打印输出的比例因子。

如果 *nPrinterSetting* 等于 6 (FOXPRO.H 中的 PRT_COPIES)，PRTINFO（）返回需要打印的副本数目。

如果 *nPrinterSetting* 等于 7 (FOXPRO.H 中的 PRT_DEFASOURCE)，PRTINFO（）返回默认的纸张来源：

返回值	设置
1	上层纸盒
2	下层纸盒
3	中层纸盒
4	人工送纸
5	信封纸盒
6	人工送纸的信封
7	自动选取
8	输纸器送纸
9	小型样式
10	大型样式
11	大容量
14	卡式盒

如果 *nPrinterSetting* 等于 8 (FOXPRO.H 中的 PRT_PRTQUAL)，PRTINFO（）返回一个正值，以每英寸的点数 (DPI) 指示水平分辨率，或一个负值指示打印质量：

返回值	设置
–1	草图
–2	低档
–3	中档
–4	高档

如果 *nPrinter* Setting 等于 9 (FOXPRO.H 中的 PRT_COLOR)，PRTINFO（）返回一个值，指示彩色打印机输出为彩色还是黑白：

返回值	设置
1	彩色
2	黑白

如果 *nPrinterSetting* 等于 10 (FOXPRO.H 中的 PRT_DUPLEX)，PRTINFO（）返回双工方式：

返回值	设置
1	单向打印
2	垂直双向打印
3	水平双向打印

如果 *nPrinterSetting* 等于 11 (FOXPRO.H 中的 PRT_YRESOLUTION)，PRTINFO（）以每英寸的点数 (DPI) 返回垂直分辨率。如果此信息不可用，返回 -1。

如果 *nPrinterSetting* 等于 12 (FOXPRO.H 中的 PRT_TTOPTION)，PRTINFO（）返回一个值，指示如何打印 TrueType 字体：

返回值	设置
1	作为位图图形打印
2	作为软字体下载
3	替代设备字体

如果 *nPrinterSetting* 等于 13，PRTINFO（）返回一个值指示输出是否排序：

返回值	设置
0	不排序
1	排序

cPrinterName

指定需要返回信息的打印机名，如果省略 *cPrinterName*，则返回默认的打印机的信息。

说明

Visual FoxPro 的打印机设置是在“页面设置”对话框中进行的。从“文件”菜单中选择“页面设置”可以显示 Visual FoxPro 的“页面设置”对话框。

请参阅

GETPRINTER(), PRINTSTATUS(), ET DEVICE, SET PRINTER, SYS(13), SYS(1037)

_PSCODE 系统变量

存放初始打印代码。可用报表设计器代替。

PSet 方法

把一个表单或 Visual FoxPro 主窗口中的一个点设置成前景色。

语法

[FormSet.]Object.PSet([nXCoord, nYCoord])

参数描述

nXCoord

按照表单 ScaleMode 属性指定的度量单位指定需要设置的点的横坐标。

nYCoord

按照表单 ScaleMode 属性指定的度量单位指定需要设置的点的纵坐标。

说明

所画点的大小由 DrawWidth 属性决定。当 DrawWidth 设置为 1 时，PSet 方法把单个像素设置成指定颜色；当 DrawWidth 设置为大于 1 的值时，点在指定坐标上居中对齐。画点的方式取决于 DrawMode 和 DrawStyle 属性的设置。

调用 PSet 方法时，CurrentX 和 CurrentY 属性设置为指定点的坐标。

应用于

Form, _SCREEN

请参阅

CurrentX, CurrentY 属性, DrawMode 属性, DrawStyle 属性, DrawWidth 属性, ForeColor 属性

_PSPACING 系统变量

存放打印机行间距。可用报表设计器代替。

PUBLIC 命令

定义全局变量或数组。

语法

PUBLIC MemVarList

–或者–

PUBLIC [ARRAY] ArrayName1(nRows1 [, nColumns1])
[, ArrayName2(nRows2 [, nColumns2])] ...

参数描述

MemVarList

指定一个或多个要初始化或指定为全局变量的变量。

[ARRAY] ArrayName1 (nRows1 [, nColumns1])
[, ArrayName2 (nRows2 [, nColumns2])] ...

指定一个或多个数组，将它们初始化或命名为全局数组。有关每个参数的说明，请参阅 DIMENSION。

说明

对于当前 Visual FoxPro 工作期中执行的任何程序，都可以使用和修改全局变量和数组。

除了初始化为“真”(.T.)的公共变量 FOX 和 FOXPRO 以外，用 PUBLIC 命令创建的变量和数组都初始化为“假”(.F.)。公共变量 FOX 和 FOXPRO 可以用来根据正在运行的用户产品的不同，有条件地执行代码。

在命令窗口中，创建的任何变量或数组都自动设置为公有。

任何想要声明为公共的变量或数组，必须在赋值之前把它声明为公有。

如果在一个程序中先给一个变量或数组赋值，然后用 PUBLIC 把它声明为公共，Visual FoxPro 将会产生语法错误。

示例

```
SET TALK OFF
PUBLIC val1,val2
val1 = 10
val2 = 15
DO down
? val1
? val2

RELEASE ALL   && 只释放私有变量
DISPLAY MEMORY LIKE val?
RELEASE val1,val2   && 明确释放公共变量
DISPLAY MEMORY LIKE val?
PROCEDURE down
PRIVATE val1
val1 = 50
val2 = 100
? val1
? val2
RETURN
```

请参阅

DIMENSION, FUNCTION, LOCAL, LPARAMETERS, PARAMETERS, PARAMETERS（）,PRIVATE, PROCEDURE ,RELEASE

PUSH KEY 命令

把所有当前的 ON KEY LABEL **命令设置压入内存中的一个堆栈。**

语法

PUSH KEY
[CLEAR]

参数描述

CLEAR

从内存中清除所有当前的键指定值。

说明

当 PUSH KEY **与** POP KEY **命令一起使用时，可以先保存用** ON KEY LABEL **命令定义的键指定值，再更改这些指定值，最后恢复为原来的指定值。**

例如，在打开一个浏览窗口时，可能想使用一系列新的 ON KEY LABEL **命令。在打**

开浏览窗口之前，应使用 PUSH KEY 命令把当前的 ON KEY LABEL 键指定值保存到内存中，这时可以针对浏览窗口添加和更改 ON KEY LABEL 指定值。在关闭浏览窗口之后，可以用 POP KEY 从内存中恢复原来的 ON KEY LABEL 键指定值。

ON KEY LABEL 设置按后进先出顺序压进和弹出堆栈。

可用 DISPLAY STATUS 和 LIST STATUS 命令显示 ON KEY LABEL 定义的当前键指定值。

请参阅

ON KEY LABEL, ON(), POP KEY

PUSH MENU 命令

把一个菜单栏定义压进内存中的菜单栏定义堆栈。

语法

PUSH MENU MenuBarName

参数描述

MenuBarName

指定要压入堆栈的菜单栏名。菜单栏可以是用户自定义菜单栏，也可以是 Visual FoxPro 系统菜单栏。

说明

当 PUSH MENU 与 POP MENU 一起使用时，可以先保存一个菜单栏定义，再修改菜单栏定义，最后恢复成初始状态。

菜单栏定义按照后进先出顺序压进和弹出堆栈。

示例

在下面 example 中，将 Visual FoxPro 系统菜单栏压进堆栈并修改此菜单栏，然后还原初始的系统菜单栏。

```
WAIT WINDOW '按下键保存系统菜单栏'
PUSH MENU _MSYSMENU
WAIT WINDOW '按下键改变系统菜单栏'
SET SYSMENU TO _MFILE, _MEDIT
WAIT WINDOW '按下键恢复系统菜单栏'
POP MENU _MSYSMENU
```

请参阅

ACTIVATE MENU, DEFINE MENU

PUSH POPUP 命令

把一个菜单定义压进内存中的菜单定义堆栈。

语法

PUSH POPUP MenuName

参数描述

MenuName

当 PUSH POPUP 与 POP POPUP 一起使用时，可以先保存一个菜单定义，然后再修改菜单定义，然后恢复成初始状态。菜单定义按照后进先出顺序压进和弹出堆栈。

说明

当 PUSH POPUP 与 POP POPUP 一起使用时，可以先保存一个菜单定义，然后再修改菜单定义，然后恢复成初始状态。

菜单定义按照后进先出顺序压进和弹出堆栈。

示例

在下列 example 中，创建了一个名为 popExam 的菜单。先将菜单定义压进堆栈，然后修改此菜单。最后，弹出堆栈，恢复初始菜单。

```
DEFINE POPUP popExam FROM 5,5
DEFINE BAR 1 OF popExam PROMPT 'One'
DEFINE BAR 2 OF popExam PROMPT 'Two'
DEFINE BAR 3 OF popExam PROMPT 'Three'
DEFINE BAR 4 OF popExam PROMPT 'Four'
ACTIVATE POPUP popExam NOWAIT
PUSH POPUP popExam
WAIT '原始菜单' WINDOW
RELEASE BAR 2 OF popExam
```

```
WAIT '将原始菜单保存到堆栈中，并修改菜单.' WINDOW
POP POPUP popExam
WAIT '恢复原始菜单' WINDOW
DEACTIVATE POPUP popExam
RELEASE POPUP popExam
```

请参阅

ACTIVATE POPUP, DEFINE POPUP

PUTFILE（）函数

激活“另存为…”对话框，并返回指定的文件名。

语法

PUTFILE([cCustomText] [, cFileName] [, cFileExtensions])

返值类型

字符型

参数描述

cCustomText

指定在“另存为…”对话框上部显示的标题。在 Windows 3.x 中，该文本显

示为对话框标题。在 Windows 95 中，该文本替换“文件名”标签。

注意：在 Windows 95 中，长标题可能被截断。

cFileName

指定在文本框中显示的默认文件名。

cFileExtensions

指定文件的扩展名。当清除“所有文件”复选框时，在“另存为 …”对话框的可滚动列表中，只显示具有指定扩展名的文件名。如果输入的文件名不包含扩展名，*cFileExtensions* 中的第一个扩展名会自动加到文件名上。

字符表达式 *cFileExtensions* 可以是下列某一形式：

- *cFileExtensions* 可以包含单个的扩展名（例如 PRG），只有带这个扩展名的文件名才被显示。
- *cFileExtensions* 可以包含用分号分隔的文件扩展名列表。例如，如果包含 PRG;FXP，Visual FoxPro 将显示所有带 PRG 和 FXP 扩展名的文件名。
- 如果文件名具有相同的基本名，但扩展名不同（例如，CUSTOMER.PRG 和 CUSTOMER.FXP)，Visual FoxPro 所显示的文件是扩展名在 *cFileExtension* 中位置靠前的文件。
- *cFileExtensions* 可包含用竖线分隔的文件扩展名列表，例如 PRG|FXP。在这种情况下，即使文件有相同的基本名，Visual FoxPro 也把所有具有指定扩展名的文件名显示出来。
- 如果 *cFileExtensions* 只包含一个分号 (;)，Visual FoxPro 显示所有不带扩展名的文件名。

- 如果 *cFileExtensions* 是个空字符串，Visual FoxPro 显示当前目录中的所有文件名。
- 如果 *cFileExtensions* 包含 MS-DOS 通配符，例如问号 (?) 和星号 (*)，Visual FoxPro 显示所有扩展名符合通配符条件的文件名。例如，如果 *cFileExtensions* 包含 ?X?，扩展名为 .FXP，.EXE，.TXT 等等的文件都会显示。

说明

可以使用 PUTFILE() 选择现有的文件名或指定新文件名，PUTFILE() 返回一个文件名及其路径。如果不输入文件名，PUTFILE() 返回默认的文件名 (由 *cFileName* 指定) 和扩展名 (由 *cFileExtensions* 指定)；如果选定“取消”或按 ESC 键，PUTFILE() 返回一个空字符串。可以使用 PUTFILE() 返回的文件名命名一个文件并把它保存到磁盘上。

示例

下面的示例从用户选择的现有表中创建一个分隔数据文件。GETFILE() 用于查找和打开表，PUTFILE() 用于返回目标文件的名称。

```
gcTableName = GETFILE('DBF', 'Open Table:')
USE (gcTableName)
gcDelimName = ALIAS() + '.DLM'
gcDelimFile = PUTFILE('Delimited file:', gcDelimName, 'DLM')
IF EMPTY(gcDelimFile)  && 按 Esc 键
    CANCEL
ENDIF
COPY TO (gcDelimFile) DELIMITED   && 创建一个界定的文件
```

MODIFY FILE (gcDelimFile) NOEDIT

请参阅

FILE( ), GETEXPR, GETFILE( ), GETPICT( ), LOCFILE( )

PV（）函数

返回某次投资的现值。

语法

PV(nPayment, nInterestRate, nTotalPayments)

返值类型

数值型

参数描述

nPayment

指定的阶段支付额。*nPayment*可取正值或负值。PV（）假定在每个阶段末支付。

nInterestRate

指定的阶段利率。如果投资的利率是按年计算而按月支付的话，把年利率除

以 12。

nTotalPayments

指定支付的总次数。

说明

在固定的阶段利率和一系列等值的阶段支付额基础上，PV（）函数计算投资的现值。

示例

```
STORE 500 to gnPayment &&阶段支付是按月进行的
STORE .075/12 TO gnInterest &&  7.5%的年利率
STORE 48 TO gnPeriods &&4 年(48 个月)
CLEAR
? PV(gnPayment, gnInterest, gnPeriods)  && 显示 20679.19
```

请参阅

CALCULATE, FV()

_PWAIT 系统变量

指定打印机输出时是否在页之间暂停。包含此变量是为了提供向后兼容性，可用报表设计器代替。

QueryAddFile 事件

在一个文件添加到项目之前发生。

语法

PROCEDURE Object.QueryAddFile
[LPARAMETERS cFileName]

参数描述

cFileName

指定添加到项目中的文件的名称。当执行 Add 方法之后，在项目管理器中选择“添加”按钮，或从“项目”菜单选择“添加文件”命令时，cFileName 会传递到 QueryAddFile 事件。

说明

在 QueryAddFile 事件中包含 NODEFAULT，或者返回“假” (.F.)，可以防止将一个文件添加到项目中。

应用于

ProjectHook 对象

请参阅

QueryModifyFile 事件

在项目中的一个文件被修改之前发生。

语法

PROCEDURE Object.QueryModifyFile
[LPARAMETERS oFile, *cClassName*]

参数描述

oFile

指定要修改文件的对象引用。当执行了 Modify 方法之后，从中选择“修改”按钮，或者从“项目”菜单中选择“修改文件”命令时，oFile 被传递给 QueryModifyFile 事件。

cClassName

包含当文件为“可视类库”.vcx 类型时，要修改的类名称。

说明

在 QueryModifyFile 事件中包含 NODEFAULT，可以防止在项目中修改一个文件。

应用于

ProjectHook 对象

请参阅

Modify 方法, QueryAddFile 事件, QueryRemoveFile 事件, QueryRunFile 事件

QueryRemoveFile 事件

当从项目中移去一个文件之前发生。

语法

PROCEDURE Object.QueryRemoveFile
[LPARAMETERS oFile, *cClassName*, lD*eleteFile*]

参数描述

oFile

指定要从项目中移去文件的对象引用。当执行了 Remove 方法之后，从项目管理器中选择“移去”按钮，或从“项目”菜单中选择“移去文件”命令之后，oFile 被传递给 QueryRemoveFile 事件。

cClassName

包含当文件为“可视类库”.vcx 类型时，要删除的类名称。

lDeleteFile

包含一个逻辑值，以表示是否此文件同项目一样，已从磁盘中删除。试图从项目中删除文件时，会显示一个对话框，如果在此对话框中选定“删除”（从项目中删除），则 lDeleteFile 包含“假”(.F.)。如果在此对话框中选定“删除”（从项目和磁盘中删除），则 lDeleteFile 包含“真”(.T.)。

说明

在 QueryRemoveFile 事件中包含 NODEFAULT，可以防止从项目中移去一个文件。

应用于

ProjectHook 对象

请参阅

QueryAddFile 事件, QueryModifyFile 事件, QueryRunFile 事件, Remove 方法

QueryRunFile 事件

当运行一个文件之前，或者在项目中预览一个报表或标签之前发生。

语法

PROCEDURE Object.QueryRunFile
[LPARAMETERS oFile]

参数描述

oFile
指定要运行或预览的文件的对象引用。在执行 Run 方法之后，从项目管理器中选择“运行”或“预览”按钮，或者从“项目”菜单选择“运行文件”或“预览文件”命令之后，oFile 被传递给 QueryRunFile 事件。

说明

在 QueryRunFile 事件中包含 NODEFAULT，可以防止运行或预览一个文件。

应用于

ProjectHook 对象

请参阅

QueryAddFile 事件, QueryModifyFile 事件, QueryRemoveFile 事件, Run 方法

QueryUnload 事件

在卸载一个表单之前发生此事件。

语法

PROCEDURE Form.QueryUnload

说明

QueryUnload 事件发生在 Destroy 事件之前。调用 QueryUnload 事件之前应先设置 ReleaseType 属性。

当在代码中执行 CLEAR WINDOWS，RELEASE WINDOWS，或 QUIT 等命令时，或者当用户双击窗口弹出菜单图标时，或者当用户从表单的窗口弹出菜单中选择执行“关闭”命令时，发生 QueryUnload 事件。

注意 当在代码中执行 RELEASE 命令或调用表单的 Release 方法时，不会发生 QueryUnload 事件。

在 QueryUnload 事件过程中执行 NODEFAULT 可以阻止表单卸载。

应用于

表单

请参阅

CLEAR, ControlBox 属性, DEFINE CLASS, Destroy 事件, QUIT, RELEASE, Release 方法, RELEASE WINDOWS, ReleaseType 属性, Unload 事件

QUIT 命令

退出 Visual FoxPro 的一个实例。

语法

QUIT

说明

调用 Quit（）方法，将控制返回给创建 Visual FoxPro 实例的应用程序。

警告 请始终使用 QUIT 来终止 Visual FoxPro 工作期。如果没有发出 QUIT 命令而关闭了计算机，那么可能在打开文件时会有危险，并且数据丢失，正常情况下已被删除的临时工作文件可能仍然保留在磁盘上。

请参阅

CANCEL, RESUME, RUN, SUSPEND

Quit 方法

退出 Visual FoxPro 的一个实例。

语法

ApplicationObject.Quit( )

说明

调用 Quit（）方法，将控制返回给创建 Visual FoxPro 实例的应用程序。

应用于

Application **对象，** _VFP **系统变量**

请参阅

QUIT

RAND（）函数

返回一个 0 到 1 之间的随机数。

语法

RAND([nSeedValue])

返值类型

数值型

参数描述

nSeedValue

指定的种子数值，它决定 RAND（）函数返回的数值序列。

在第一次发出 RAND（）函数时使用一个种子数值 *nSeedValue* 后，以后再使用 RAND（）函数时不带 *nSeedValue*，将得到一个随机数序列。如果第三次发出 RAND（）函数时使用同样的种子数值 *nSeedValue*，那么 RAND( )返回同样的随机数序列。

如果第一次发出 RAND（）时使用的 *nSeedValue* 参数是负数，那么将使用来自系统时钟的种子值。若要获得随机程度最大的数字序列，可以最初用一个负的参数发出 RAND（）函数，然后再不带参数发出 RAND（）函数。

如果省略种子数值，RAND（）函数将采用默认值 100,001。

示例

下面的第一个示例用 RAND（）创建了一个包含随机值的 10 个记录的表，再用 MIN（）和 MAX（）来显示表中的最大值和最小值。

第二个示例显示了在两个值 1 和 10 之间的一个随机值。

```
CLOSE DATABASES
CREATE TABLE Random (cValue N(3))
FOR nItem = 1 TO 10  && 增加 10 个记录
    APPEND BLANK
    REPLACE cValue WITH 1 + 100 * RAND（）  && 插入任意值
ENDFOR

CLEAR
LIST  && 显示这些值
gnMaximum = 1  && 初始化最小值
gnMinimum = 100  && 初始化最大值
SCAN
    gnMinimum = MIN(gnMinimum, cValue)
    gnMaximum = MAX(gnMaximum, cValue)
ENDSCAN
? 'The minimum value is: ', gnMinimum  && 显示最小值
? 'The maximum value is: ', gnMaximum  && 显示最大值
CLEAR
gnLower = 1
gnUpper = 10

? INT((gnUpper - gnLower + 1) * RAND（） + gnLower)
```

请参阅

EXP( ), PI( )

RangeHigh 事件

对于微调控件或文本框，当控件失去焦点时，此事件发生。对于组合框或列表框，当控件得到焦点时，此事件发生。

语法

PROCEDURE Object.RangeHigh
[LPARAMETERS nIndex]

参数描述

nIndex

唯一标识位于控件数组中的一个控件。

说明

RangeHigh 事件能够通过 RETURN 语句返回给 Visual FoxPro 一个数值。对于具有某个数值的微调控件或文本框，如果返回给 Visual FoxPro 的数值小于输入到控件中的数值，那么该控件仍然保持焦点。对于组合框或列表框，返回给 Visual FoxPro 的数值指

定初始时选定控件中的哪一个数据项。例如，如果 2 返回给 Visual FoxPro，那么初始时选定控件中的第二个数据项。

应用于

组合框，列表框，微调，文本框

请参阅

KeyboardHighValue, KeyboardLowValue 属性, RangeLow 事件, RETURN, SpinnerHighValue, SpinnerLowValue 属性

RangeLow 事件

对于微调控件或文本框，当控件失去焦点时，此事件发生；对于组合框或列表框，当控件获得焦点时，发生此事件。

语法

PROCEDURE Object.RangeLow
[LPARAMETERS nIndex]

参数描述

nIndex

唯一标识位于控件数组中的一个控件。

说明

Range Low 事件能够通过 RETURN 语句返回给 Visual FoxPro 一个数值。对于具有某个数值的微调控件或文本框，如果返回给 Visual FoxPro 的数值大于输入到控件中的数值，那么该控件仍然保持焦点。对于组合框或列表框，返回给 Visual FoxPro 的数值指定初始时选定控件中的哪一个数据项。例如，如果 2 返回给 Visual FoxPro，那么初始时选定控件中的第二个数据项。

应用于

组合框，列表框，微调，文本框

请参阅

KeyboardHighValue, KeyboardLowValue 属性, RangeHigh 事件, RETURN, SpinnerHighValue, SpinnerLowValue 属性

RAT（）函数

返回字符表达式或备注字段在另一个字符表达式或备注字段内第一次出现的位置，从最右边的字符算起。

语法

RAT(cSearchExpression, cExpressionSearched [, nOccurrence])

返值类型

数值型

参数描述

cSearchExpression

指定 RAT（）函数要在 *cExpressionSearched* 中搜索的字符表达式。

cExpressionSearched

指定 RAT（）函数搜索的字符表达式。字符表达式 *cSearchExpression* 和 *cExpressionSearched* 可以是任意大小的备注字段。

nOccurrence

指定 RAT（）在 *cExpressionSearched* 中从右到左搜索 *cSearchExpression* 的第几次出现时的位置。默认时，RAT（）函数搜索 *cSearchExpression* 的最后一次出现 (*nOccurrence*=1) 时的位置。如果 *nOccurrence* 等于 2，RAT（）函数将搜索倒数第二次出现的位置，依此类推。

说明

与 AT（）函数相反，RAT（）函数在字符表达式 *cExpressionSearched* 中从右到左搜索另一个字符表达式 *cSearchExpression* 的最近一次出现的位置。

RAT（）函数返回一个整数，该整数指示 *cSearchExpression* 的第一个字符在

cExpressionSearched 中的位置。如果在 *cExpressionSearched* 中没有找到 *cSearchExpression*，或者 *nOccurrence* 大于 *cSearchExpression* 在 *cExpressionSearched* 中的出现次数，那么 RAT（）函数返回 0。

RAT（）函数执行的搜索区分大小写。

示例

```
STORE 'abracadabra' TO string
STORE 'a' TO find_str
CLEAR
? RAT(find_str,string)  && 显示 11
? RAT(find_str,string,3)  && 显示 6
```

请参阅

A T ( ), ATCLINE( ), ATLINE( ), LEFT( ), OCCURS( ), RATLINE( ), RIGHT( ), SUBSTR( )

RATC（）函数

返回字符表达式或备注字段在另一个字符表达式或备注字段内第一次出现的位置，从最右边的字符算起。

语法

RATC(cSearchExpression, cExpressionSearched [, nOccurrence])

返值类型

数值型

参数描述

cSearchExpression

指定 RATC() 函数要在 *cExpressionSearched* 中搜索的字符表达式。

cExpressionSearched

指定 RATC() 函数搜索的字符表达式。字符表达式 *cSearchExpression* 和 *cExpressionSearched* 可以是任意大小的备注字段。

nOccurrence

指定 RATC() 在 *cExpressionSearched* 中从右到左搜索 *cSearchExpression* 的第几次出现时的位置。默认时，RATC() 函数搜索 *cSearchExpression* 的最后一次出现 (*nOccurrence*=1) 时的位置。如果 *nOccurrence* 等于 2，RATC() 函数将搜索倒数第二次出现的位置，依此类推。

说明

RATC() 设计用于处理包含双字节字符的表达式。如果表达式中只有单字节字符，则 RATC() 等同于 RAT()。

RATC() 返回第一个字符表达式或备注字段在第二个字符表达式或备注字段中最后

一次出现的数值位置，字符表达式可以包含单字节和双字节字符的任意组合。
RATC（）与 AT_C（）函数相反：它从右到左搜索。
RATC（）函数返回一个整数，该整数指示 *cSearchExpression* 的第一个字符在 *cExpressionSearched* 中的位置。如果在 *cExpressionSearched* 中没有找到 *cSearchExpression*，或者 *nOccurrence* 大于 *cSearchExpression* 在 *cExpressionSearched* 中的出现次数，那么 RATC（）函数返回 0。
RATC（）函数执行的搜索区分大小写。
该函数适用于管理双字节字符集。

请参阅

A T _ C ( ), LEFTC( ), RAT( ), RIGHTC( ), SUBSTRC( )

RATLINE（）函数

返回字符表达式或备注字段在另一个字符表达式或备注字段中最后一次出现的行号，从最后一行开始计数。

语法

RATLINE(cSearchExpression, cExpressionSearched)

返值类型

数值型

参数描述

cSearchExpression

指定 RATLINE() 函数要在 *cExpressionSearched* 中搜索的字符表达式。

cExpressionSearched

指定 RATLINE() 函数搜索的字符表达式。字符表达式 *cSearchExpression* 和 *cExpressionSearched* 可以是任意大小的备注字段。

使用 MLINE() 函数可以返回包含 cSearchExpression 的行。

提示 RATLINE() 函数为搜索备注字段提供了一个简便的方法。

说明

与 RATLINE() 函数相反，RATLINE() 函数从 *cExpressionSearched* 中的最后一个字符开始，在字符表达式 *cExpressionSearched* 中搜索 *cSearchExpression* 出现的行号。如果搜索成功，RATLINE() 函数返回匹配的行号；如果失败，RATLINE()返回 0。RATLINE() 执行的搜索区分大小写。

注意 即使 *cExpressionSearched* 不是备注字段，RATLINE() 函数所返回的行号也由 SET MEMOWIDTH 的值决定。

示例

下面的示例中，RATLINE() 返回了在 notes 备注字段中包含单词 “graduated.” 的最后一行的行号。MLINE() 用该值返回行的内容。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'data\testdata')
```

```
USE employee  && 打开 Employee 表
STORE 'graduated' TO gcString
STORE MLINE(notes, RATLINE(gcString, notes)) TO gnFileLine
? gnFileLine
```

请参阅

A T ( ), ATCLINE( ), ATLINE( ), LEFT( ), MLINE( ), OCCURS( ), RAT( ), RIGHT( ), SUBSTR( )

RD | RMDIR 命令

从磁盘上删除一个目录。

语法

RD cPath | RMDIR cPath

参数描述

cPath

指定要从磁盘上删除的目录名及其位置。

说明

如果试图删除一个非空的目录， Visual FoxPro **将产生错误信息。**

示例

下面的示例用 MKDIR 创建了一个新目录 mytstdir，使用 CHDIR 转到新目录。GETDIR() 用于显示目录结构，然后用 RMDIR 删除新建目录。再用 GETDIR() 来显示目录结构。

```
SET DEFAULT TO HOME() && 恢复 Visual FoxPro 目录
MKDIR mytstdir && 创建一个新目录
CHDIR mytstdir && 转换到新目录下
= GETDIR() && 显示选择目录对话框
SET DEFAULT TO HOME() && 恢复 Visual FoxPro 目录
RMDIR mytstdir && 删除新目录
= GETDIR() && 显示选择目录对话框
```

请参阅

CD | CHDIR, GETDIR(), HOME(), MD | MKDIR, SET DEFAULT, SET PATH, SYS(5), SYS(2003), SYS(2004)

RDLEVEL() 函数

包含此项是为了提供向后兼容性。请使用表单设计器。

READ EVENTS 命令

开始事件处理。

语法

READ 事件

说明

当使用 READ EVENTS 命令时，Visual FoxPro 开始事件处理。

发出 CLEAR EVENTS 命令，可以终止事件处理。当发出 CLEAR EVENTS 命令时，程序在 READ EVENTS 紧后面一行继续执行。

注意，一次只能有一个 READ EVENTS 命令是活动的。如果一个 READ EVENTS 是活动的，后面的任何 READ EVENTS 命令都不起作用。

请参阅

CLEAR 命令，Form Designer

READ 命令

包含此命令是为了提供向后兼容性，可用表单设计器代替。

READ MENU 命令

包含此命令是为了提供向后兼容性，使用菜单设计器可以创建菜单。

ReadActivate 事件

包含此事件是为了 READ 命令的向后兼容性。请使用表单设计器。

adBackColor, ReadForeColor 属性

READ 命令的向后兼容性。请使用表单设计器。

ReadCycle 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadDeactivate 事件

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadExpression 方法

返回属性窗口中某属性的表达式。仅在设计时可用。

语法

Object.ReadExpression(cPropertyName)

参数描述

cPropertyName

指定属性的名称，从该属性中返回表达式。

应用于

ActiveDoc 对象，复选框，列，组合框，命令按钮，命令组，自定义，编辑框，表单，表单集，表格，标头，图像，标签，线条，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，选项组，页面，页框，ProjectHook 对象，形状，微调，文本框，计时器，工具栏

请参阅

WriteExpression 方法

READKEY（）函数

包含该函数是为了提供与 READ 的向后兼容性。请使用表单设计器。

ReadLock 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadMethod 方法

返回指定的方法的文本。设计和运行时可用。

语法

Control.ReadMethod(cMethod)

参数描述

cMethod
指定方法的名称。

应用于

ActiveDoc 对象，复选框，列，组合框，命令按钮，命令组，临时表，自定义，数据环境，编辑框，表单，表单集，表格，标头，图像，标签，线条，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，选项组，页面，页框，ProjectHook 对象，关系，形状，微调，文本框，计时器，工具栏

请参阅

ReadExpression 方法, WriteExpression 方法, WriteMethod 方法

ReadMouse 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadObject 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadOnly 属性

指定用户是否可以编辑一个控制，更新与临时表对象相关联的表或视图，或者包含一个表明项目中某文件是否可以编辑的值。设计时可用，运行时可读写。

语法

[Form.]Control.ReadOnly[= lExpr]
DataEnvironment.Cursor.ReadOnly[= lExpr]
Project.Files.Item(nItem).ReadOnly

设置

lExpr

下表列出了 Read Only 属性的设置：

设置	说明
“真” (.T.)	用户不能编辑控制。不能修改与临时表对象相关联的表或视图。
“假” (.F.)	（默认值）用户能够编辑控制。可以修改与临时表对象相关联的表或视图。

说明

注意 当用 CURSORSETPROP( )函数访问临时表时，ReadOnly 属性在运行时只读。

ReadOnly 属性不同于 Enabled 属性。因为当 ReadOnly 设置为“真” (.T.)时，用户仍然可以移动到控制上。

对于临时表，ReadOnly 属性与 USE 命令的 NOUPDATE 子句作用类似。

对于组合框控件，当 Style 属性设置为 2 – 下拉式列表时，ReadOnly 属性不能设置为“真” (.T.)。

对于项目中的一个文件，ReadOnly 属性包含一个逻辑值，表明是否可以编辑该文件。如果 ReadOnly 为“真” (.T.)，则可以编辑该文件；否则不可以编辑该文件。设计和运行时只读。

应用于

复选框，列，临时表，编辑框，文件对象，表格，微调，文本框

请参阅

Enabled 属性, USE

ReadSave 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadShow 事件

包含此事件是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadTimeout 属性

包含此属性是为了 READ 命令的向后兼容性。请使用表单设计器。

ReadValid 事件

此事件是提供 READ 命令的向后兼容性。请使用表单设计器。

ReadWhen 事件

包含此事件是为了提供与 READ 命令的向后兼容性。请使用表单设计器。

RECALL 命令

恢复所选表中带有删除标记的记录。

语法

RECALL

[Scope] [FOR lExpression1] [WHILE lExpression2]
[NOOPTIMIZE]

参数描述

Scope

指定要恢复记录的范围。只有在指定范围内的记录才被恢复。Scope**子句有** ALL、NEXT *nRecords*、RECORD *nRecordNumber* **和** REST**。有关** scope**子句的详细内容，请参阅帮助中的“**Scope Clauses**”和“语言概述”。**

RECALL **命令默认的范围是当前记录** (NEXT1)**。**

FOR lExpression1

指定只恢复 *lExpression1* **计算为“真”** (.T.)**的记录，这可以筛选掉不需要的记录。**

如果 *lExpression1* **是一个可优化表达式，那么** Rushmore **将优化** RECALL FOR **语句。为了得到最佳的性能，请在** FOR **子句中使用可优化表达式。**

详细内容，请参阅稍后的“语言参考”中的 SET OPTIMIZE **命令与《**Microsoft Visual FoxPro 6.0 **中文版程序员指南》第十五章“优化应用程序”中的“掌握** Rushmore **技术”。**

WHILE lExpression2

指定一个条件，只要 *lExpression2* **计算为“真”** (.T.)**时，就恢复删除的记录。**

NOOPTIMIZE

关闭 RECALL **命令的** Rushmore **优化。**

详细内容，请参阅稍后的“语言参考”中的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

说明

只要没有发出 PACK 或 ZAP 命令，就可以使用 RECALL 命令恢复记录。

警告 一旦对文件使用了 PACK 或 ZAP 命令，那么所有带删除标记的记录将永远消失。

可以通过发出 DELETE 或 DELETE-SQL 命令给记录做删除标记，或者当浏览窗口或编辑窗口活动时，从“表”菜单中选择“删除记录”菜单项。可以通过发出 RECALL 命令恢复记录，或者当浏览窗口或编辑窗口是活动时，从“表”菜单中选择“恢复记录”菜单项。

示例

下面的示例打开了数据库 testdata 中的 customer 表。用 DELETE – SQL 在 country 字段包含 USA 的地方标记所有的记录为删除。显示所有标记为删除的记录。用 RECALL ALL 为标记为删除的记录解除标记。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'data\testdata')
USE customer  && 打开 Customer 表

DELETE FROM customer WHERE country = 'USA'  && 标记删除
CLEAR
LIST FIELDS company, country FOR DELETED（） && 列出标记记录
```

RECALL ALL && 取消标记删除记录的标记

请参阅

DELETE, DELETE – SQL, PACK, SET DELETED, ZAP

RECCOUNT() 函数

返回当前或指定表中的记录数目。

语法

RECCOUNT([nWorkArea | cTableAlias])

返值类型

数值型

参数描述

nWorkArea

指定表所在的工作区编号。

如果在指定的工作区中没有打开的表，RECCOUNT() 返回 0。

cTableAlias

指定表别名。

说明

SET DELETED 和 SET FILTER 命令并不影响 RECCOUNT() 函数的返回值。

不带可选参数 *nWorkArea* 或 *cTableAlias* 的 RECCOUNT() 函数返回当前所选工作区中表的记录数目。

示例

在下面的 example 中，Visual FoxPro 将 customer 表排序所需要的空间与可用的磁盘空间进行比较。

```
*** 在 SORT 命令之前检查磁盘空间 ***

CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
USE customer  &&打开 customer 表

*** 获得表头大小 r ***

gnTableHead = HEADER( )

*** 计算表的大小 ***

gnFileSize = gnTableHead + (RECSIZE() * RECCOUNT() + 1)
IF DISKSPACE() > (gnFileSize * 3)
    WAIT WINDOW 'Sufficient diskspace to sort.'
ELSE
    WAIT WINDOW 'Insufficient diskspace. Sort cannot be done.'
ENDIF
```

请参阅

RECNO( ), RECSIZE( )

RECNO（）函数

返回当前表或指定表中的当前记录号。

语法

RECNO([nWorkArea | cTableAlias])

返值类型

数值型

参数描述

nWorkArea

指定表所在的工作区编号。如果在指定的工作区中没有打开的表，RECNO（）函数返回 0。

cTableAlias

指定表别名。

说明

当前记录就是记录指针所指的记录。

对于在表缓冲区中追加的记录，RECNO（）返回负记录编号。

如果记录指针所指的位置超出了表中的最后一个记录，那么RECNO（）函数返回一个比表中记录数目大1的数值；如果记录指针所指的位置在表中第一个记录之前或者表中没有记录，那么RECNO（）函数返回1；如果表中没有记录，EOF（）函数总是返回“真”(.T.)。

执行不带可选参数 *nWorkArea* 或 *cTableAlias* 的RECNO（）函数，将返回当前所选工作区中表的当前记录号。

如果在一个经过索引的表中发出SEEK命令失败，那么可以将 *nWorkArea* 指定为0，使用“软寻找(softseek)”方法来返回最接近匹配记录的记录号；如果不能找到接近的匹配记录，RECNO(0)函数返回0；如果没有找到接近的匹配记录而发出GO RECNO(0)命令，那么Visual FoxPro将产生错误信息。

示例

下面的example在customer表中搜索一个公司的名称。如果没有找到该公司的名称，就使用RECNO(0)返回最接近匹配的公司名称。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'data\testdata')
USE customer  && 打开 customer 表
SET ORDER TO company
SEEK 'Ernst'
IF FOUND( )
    DISPLAY company, contact
ELSE
    GOTO RECNO(0)
    CLEAR
```

```
    ? 'Closest matching company is ' + company
    ? 'Record number: ' + ALLTRIM(STR(RECNO( )))
ENDIF
```

请参阅

GO | GOTO, RECCOUNT( ), RECSIZE( ) , SEEK, SKIP

RecordMark 属性

指定是否在表格控制中显示记录选择器一列。设计时可用，运行时可读写。

语法

Grid.RecordMark[= lExpr]

参数描述

lExpr

下表列出了 RecordMark 属性的设置：

设置	说明
“真” (.T.)	（默认值）在表格控制中显示记录选择器列。

续表

“假”(.F.)	在表格控制中不显示记录选择器列。

应用于

表格

请参阅

ScrollBars 属性

RecordSource 属性

指定与表格控制相绑定的数据源。设计时可用，运行时只读写。

语法

Grid.RecordSource[= cName]

参数描述

cName

cName 一般是临时表的别名或者表的名称。RecordSource 属性指定与表格控

制相绑定的主临时表。

说明

如果为一个网格指定了记录源，就可以通过设置 ControlSource 属性来指定网格中单独列的内容。如果没有为网格中的列设置 ControlSource 属性，那么列显示网格的记录源中的下一个可用的不能显示的字段。

应用于

表格

请参阅

ControlSource 属性, Order 属性, RecordMark 属性, RecordSourceType 属性

RecordSourceType 属性

指定如何打开填充表格控制的数据源。设计时可用，运行时可读写。

语法

Grid.RecordSourceType[= nType]

参数描述

nType

RecordSourceType 属性的设置有：

设置	说明
0	表。自动打开 RecordSource 属性设置中指定的表。
1	(默认值)别名。按指定方式处理记录源。
2	提示。在运行时向用户提示记录源。如果某个数据库已打开，用户可以选择其中的一个表作为记录源。
3	查询(.QPR)。RecordSource 属性设置指定一个 .QPR 文件。
4	SQL 语句。在 RecordSource 属性中指定 SQL 语句。

应用于

表格

请参阅

RecordSource 属性

RECSIZE（）函数

返回表中记录的大小（宽度）。

语法

RECSIZE([nWorkArea | cTableAlias])

返值类型

数值型

参数描述

nWorkArea

指定表所在的工作区编号。如果指定工作区中没有打开的表，RECSIZE( )返回 0。

cTableAlias

指定非当前工作区中打开表的别名。

说明

不带可选参数 *nWorkArea* 和 *cTableAlias* 发出 RECSIZE（）函数，将返回当前选定工作区中表的记录大小。

示例

在下面的 example 中，Visual FoxPro 比较可用的磁盘空间与对 customer 表排序所需空间的大小。

```
***在排序前检查磁盘空间 ***

CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
USE customer && 打开表 Customer
***得到表头的大小 ***

gnTableHead = HEADER( )

***计算表的大小 ***

gnFileSize = gnTableHead + (RECSIZE ( ) * RECCOUNT ( ) + 1)
IF DISKSPACE ( ) > (gnFileSize * 3)
    WAIT WINDOW 'Sufficient diskspace to sort.'
ELSE
    WAIT WINDOW 'Insufficient diskspace. Sort cannot be done.'
ENDIF
```

请参阅

RECCOUNT( ), FSIZE( )

Refresh 方法

重画表单或控件，并刷新所有值，或者刷新一个项目的显示。

语法

[Form.]Object.Refresh

– 或者 –

Project.Refresh([lUpdateSCCStatus])

参数描述

lUpdateSCCStatus

对于一个项目，指定是否更新源代码管理器中的每个文件的源代码管理状态。如果 lUpdateSCCStatus 为“真” (.T.)，则更新源代码管理器中的每个文件的源代码管理状态。如果 lUpdateSCCStatus 为“假” (.F.) 或省略，则不更新源代码管理器中的每个文件。如果项目中没有文件处于源代码管理器中，则忽略 lUpdateSCCStatus。

说明

一般地，画表单或控件是在没有事件发生时自动处理的。需要立刻更新表单或控件时可使用 Refresh 方法。

可使用 Refresh 方法强制地完全重画表单或控件，并更新控件的值。若要在加载另一个表单的同时显示某个表单，或更新控件的内容时，Refresh 方法很有用。要更新组合框或列表框的内容，请使用 Requery 方法。

注意 刷新表单的同时，也刷新表单上所有的控件；刷新页框时，只刷新活动的页。

对于项目，执行 Refresh 方法，会根据对该项目进行的任何更改刷新该项目的可视显示。例如，使用 Add 方法通过编程向项目中添加文件之后，可以执行 Refresh 方法，以便显示项目中新加的文件。

应用于

复选框，列，组合框，命令按钮，命令组，容器对象，控件对象，编辑框，表单，表单集，表格，标头，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，选项组，页面，页框，项目对象，_SCREEN，微调，文本框，工具栏

请参阅

Paint 事件，Requery 方法

REFRESH（）函数

在可更新的 SQL 视图中刷新数据。

语法

REFRESH([nRecords [, nRecordOffset]] [, cTableAlias | nWorkArea])

返值类型

数值型

参数描述

nRecords

指定要刷新的记录数目。如果 *nRecords* 等于 0 或省略 *nRecords*，则只刷新当前记录。

nRecordOffset

指定开始刷新的记录与当前记录的偏移记录数。例如，如果当前记录是 10 号记录，而 *nRecordOffset* 等于 -4，则从 6 号记录开始刷新；如果 *nRecordOffset* 等于 0，或省略 *nRecordOffset*，则只刷新当前记录。

cTableAlias

指定要刷新记录的远程 SQL 视图的别名。

nWorkArea

指定将在其中刷新记录的表或临时表所在的工作区。如果省略 *nWorkArea* 和 *cTableAlias*，则在当前选定工作区的远程 SQL 视图中刷新记录。

说明

REFRESH（）函数返回刷新的记录数目。

刷新记录使用的是用来创建 SQL 视图的表的数据。记录在当前选定工作区打开的 SQL 视图中刷新。

REFRESH（）不能刷新锁定或缓冲的记录，并且记录的主关键字值必须唯一。如果表中的某个记录没有主关键字，则 SQL 视图中相应的记录被做上删除标记。

提示 调用 REFRESH（）对性能有显著的影响，因为此函数重新执行了视图作为基础的查询。因此，请在必要时才调用此函数。

请参阅

CREATE SQL VIEW, CURSORGETPROP( ), CURSORSETPROP( )

REGIONAL 命令

创建局域变量和数组。

语法

#REGION nRegionNumber
REGIONAL VarList

参数描述

#REGION nRegionNumber

创建一个局域。在程序中使用局域变量之前，必须先予以声明。要注意的是 #REGION 是一条编译指令，而不是一条命令。*nRegionNumber* 指定局域号，取值范围为 0 到 31。

REGIONAL VarList

声明 #REGION 指令所创建局域中的变量。*VarList* 是由逗号分隔的变量及数组的列表。

在程序的编译过程中，如果一个局域变量已做编译，而又遇到另一个同名的局域声明时，将使第二个出现的变量名唯一，以确保不与先前已声明的局域变量发生冲突。使变量名唯一的办法是用下划线和当前局域号填充局域变量名，使之达到 10 个字符。

这种替代完全发生在程序编译过程中，对执行速度没有影响。

变量名修改后，可以使用 DISPLAY MEMORY 命令显示修改后的变量名。为了在调试窗口中监控变量，应使用修改后的变量名。由于跟踪窗口使用原始的程序源代码，因而原始变量名（而不是由编译器创建的修改后的名称）出现在跟踪窗口中。

说明

如果在局域中受保护，即使同名的变量或数组也不互相干扰。局域变量与私有变量相似。

示例

下面的示例创建了两套局部变量。在 region 1 中，创建了变量 gcA、gcB、gcC 和 gcD，并都赋值“One”字符串。在 region 2 中，创建了变量 gcC、gcD、gcE 和 gcF，并都赋值“Two”字符串。变量 gcC 和 gcD 在两个区公用。

然后显示 DISPLAY MEMORY 的输出。在第二个区中修改变量 gcC 和 gcD 的名称。gcC 变为 GCC_____2，并且 gcD 变为 GCD_____2。所有的变量都是私有的，可被低级程序访问。

```
#REGION 1
REGIONAL gcA,gcB,gcC,gcD
STORE 'One' to gcA,gcB,cgC,gcD
#REGION 2
REGIONAL gcC,gcD,gcE,gcF   && gcC 和 gcD 对两个地区通用
STORE 'Two' to gcC,gcD,gcE,gcF
DO showmemory

PROCEDURE showmemory
```

DISPLAY MEMORY LIKE g*

请参阅

PRIVATE, PUBLIC, STORE

REINDEX 命令

重建打开的索引文件。

语法

REINDEX [COMPACT]

参数描述

COMPACT

将普通的单索引 (.IDX) 文件转换为压缩的 .IDX 文件。

说明

当打开表而不打开相应的索引文件并更改索引文件的关键字段后，索引文件就变得过时。如果索引文件已过时，可以重建索引来更新这些索引文件。

REINDEX 命令处理选定工作区中打开的所有索引文件。Visual FoxPro 识别每种索引文件的类别（复合索引 (.CDX) 文件、结构 .CDX 文件及单索引（.IDX) 文件）并分别

重建索引。此命令更新 .CDX 文件中的所有标识，并且更新与表一起自动打开的结构 .CDX 文件。

对使用包含 UNIQUE 关键字的 INDEX 命令或 SET UNIQUE ON 命令创建的索引文件，在重建索引时，仍保持 UNIQUE 状态。

要重建过时的索引文件，可以执行下列命令：

```
USE TableName INDEX OutdatedIndexNames
REINDEX
```

示例

下面的示例中，ISEXCLUSIVE() 验证了 customer 表是以独占的方式打开的。因为当前工作区中的索引不是以独占的方式打开的，所以没有对表做重新索引。

```
cExclusive = SET('EXCLUSIVE')
SET EXCLUSIVE OFF
SET PATH TO (HOME(2) + 'Data\')
OPEN DATA testdata  && 打开 test 数据库
USE Customer    && 不是以独占方式打开
USE Employee IN 0 EXCLUSIVE   && 在另一个工作区以独占方式打开

IF ISEXCLUSIVE( )
   REINDEX && 只能在独占方式下完成操作
ELSE
    WAIT WINDOW 'The table has to be exclusively opened'
ENDIF

SET EXCLUSIVE &cExclusive
```

请参阅

INDEX, SET INDEX, SET EXCLUSIVE, SET UNIQUE, SYS() 函数概览, USE

Relation 对象

当在数据环境设计器中为表单、表单集或报表建立关系时创建。

语法

Relation

说明

Relation 对象允许确定或指定两个表如何相关联。

要注意的是，运行时设置 Relation 对象的属性将产生错误。要使新的属性设置起作用，必须对数据环境调用 Close Tables 及 OpenTables 方法。

有关表单和表单集的数据环境的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第九章“创建表单”中的“设置数据环境”。

有关报表的数据环境的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十二章“添加查询和报表”。

属性

Application　ChildAlias　ChildOrder
Comment　Name　OneToMany
ParentAlias　RelationalExpr　Tag

事件

Destroy　Error　Init

方法

AddProperty　ReadExpression　ReadMethod
ResetToDefault　WriteExpression

请参阅

Cursor **对象**, DataEnvironment **对象**

RELATION（） **函数**

返回为给定工作区中打开的表所指定的关系表达式。

语法

RELATION(nRelationNumber [, nWorkArea | cTableAlias])

返值类型

字符型

参数描述

nRelationNumber

指定返回哪一个关系。例如，如果 *nRelationNumber* 等于 3，RELATION（）函数返回所创建的第三个关系的关系表达式。

nWorkArea

指定表所在的工作区。如果指定工作区中没有表打开，RELATION（）函数返回空字符串。

cTableAlias

指定表别名。

说明

默认情况下，RELATION（）返回当前选定工作区中表的关系表达式。如果不存在关系，则返回空字符串。有关在表间创建关系的其他内容，请参阅 SET RELATION。可用 DISPLAY STATUS 命令与 LIST STATUS 命令显示关系表达式。发出 MODIFY DATABASE 命令将显示数据库设计器，它允许您查看并修改当前打开的数据库中各表之间的关系。发出 SET 命令将显示查看窗口，它允许您查看并修改自由表之间的关系。

示例

在下面的 example 中，打开表并设置次序，然后用 RELATION() 显示关系。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
USE customer IN 0 ORDER cust_id  && 打开表 Customer
USE employee IN 0 ORDER emp_id  && 打开表 Customer
USE orders IN  0 ORDER order_id  && 打开表 Customer
SELECT orders
SET RELATION TO emp_id INTO employee
SET RELATION TO cust_id INTO customer ADDITIVE
?RELATION(1)  && 显示 CUST_ID
?RELATION(2)  && 显示 EMP_ID
?RELATION(3)  &&显示空串
```

请参阅

DISPLAY STATUS, MODIFY DATABASE, LIST STATUS, SET, SET RELATION, SET RELATION OFF, TARGET()

RelationalExpr 属性

指定一基于父表字段的表达式，该表达式与子表中连接父、子表的索引相关。对于表格控制，设计时可用，运行时可读写。

语法

Object.RelationalExpr[= cExpr]

参数描述

cExpr

可以指定任何的 Visual FoxPro 表达式，一般是与 ChildOrder 属性指定的子表当前索引相匹配的表达式。

应用于

表格，关系

请参阅

ChildOrder 属性, LinkMaster 属性, SET RELATION

RelativeColumn 属性

指出在表格控制可见部分中的活动列。设计时不可用，运行时只读。

语法

Grid.RelativeColumn[= nColumn]

参数描述

nColumn

指定表格控制中的活动列。

说明

可以使用 RelativeColumn 属性确定活动列在表格中的相对位置。例如，如果在表格中滚动，使第一列不再可见，而活动列是表格中第一个可见列，这时 RelativeColumn 就为 1。要确定表格中活动列的绝对位置，应使用 ActiveColumn 属性。

应用于

表格

请参阅

ActiveColumn 属性, ActiveRow 属性, RelativeRow 属性

RelativeRow 属性

指出在表格控制可见部分中的活动行。设计时不可用，运行时只读。

语法

Grid.RelativeRow[= nRow]

参数描述

nRow

指定表格控制中的活动行。

说明

可以使用 RelativeRow 属性确定活动行在表格中的相对设置。例如，如果在表格中滚动，使第一行不再可见，而活动行是表格中第一个可见行，这时 RelativeRow 就为 1。要确定表格中活动行的绝对位置，应使用 Active Row 属性。

应用于

表格

请参阅

ActiveColumn 属性, ActiveRow 属性, RelativeColumn 属性

RELEASE 命令

从内存中删除变量和数组。

语法

RELEASE MemVarList

–或者–

RELEASE ALL [EXTENDED]

– 或者 –

RELEASE ALL [LIKE Skeleton | EXCEPT Skeleton]

参数描述

RELEASE MemVarList

指定要从内存中释放的变量和数组。各个变量可用逗号分隔。

RELEASE ALL

从内存中释放所有的变量和数组。

EXTENDED

从程序中发出时，指定释放所有的公共变量。当在程序中执行 RELEASE ALL **时，并不释放公共变量。**

LIKE Skeleton | EXCEPT Skeleton

从内存中释放所有与指定梗概相匹配的变量和数组，或释放与指定梗概相匹配的变量和数组之外的所有变量和数组。 *Skeleton* **可以包含通配符 ?和 *。当在程序中执行** RELEASE ALL LIKE **或** RELEASE ALL EXCEPT **时，并不释放公共变量。**

请参阅

CLEAR MEMORY, DIMENSION, PRIVATE, PUBLIC, STORE

RELEASE BAR 命令

从内存中删除菜单上的指定菜单项或所有菜单项。

语法

RELEASE BAR nMenuItemNumber OF MenuName | ALL OF MenuName

参数描述

nMenuItemNumber OF MenuName

指定要从内存中删除的用户自定义菜单上的某个菜单项。要从 Visual FoxPro 系统菜单上删除菜单项，应在 *nMenuItemNumber* 中包含 Visual FoxPro 内部的系统菜单项名称。

ALL OF MenuName

指定从内存中删除用户自定义菜单上的每个菜单项。从 Visual FoxPro 系统菜单上删除菜单项时不能使用 ALL 子句。

请参阅

DEFINE BAR, RELEASE PAD

RELEASE CLASSLIB 命令

关闭包含类定义的 .VCX 可视类库。

语法

RELEASE CLASSLIB ClassLibraryName1 | ALIAS AliasName1
[, ClassLibraryName2 | ALIAS AliasName2 ...]
[IN APPFileName | EXEFileName]

参数描述

ClassLibraryName1 | ALIAS AliasName1
指定要关闭的可视类库文件名称。

ClassLibraryName2 | ALIAS AliasName2 ...
指定了附加可视类库文件的名称或别名以关闭。

IN APPFileName | EXEFileName
指定了一个 Visual FoxPro 应用程序文件 (.app) 或包含可视类库的执行文件 (.exe)。

说明

可以使用 SET CLASSLIB 命令打开 .VCX 可视类库。一旦打开后，可视类库中的类定义即可在程序和命令窗口中使用。

要关闭所有打开的可视类库，应发出不带任何附加参数的 SET CLASSLIB TO 命令。

示例

下面的 example 先使用 CREATE CLASSLIB 命令创建一个名为 myclslib 的可视类库，然后创建一个基于 Visual FoxPro 中 Form 基类、命名为 myform 的类，并将该类存储在 myclslib 可视类库中。先使用 SET CLASSLIB 命令打开 myclslib 可视类库，然后再用 RELEASE CLASSLIB 命令关闭可视类库 myclslib。

```
CREATE CLASSLIB myclslib   && 创建一个新的 .VCX 可视类库
CREATE CLASS myform OF myclslib AS "Form"  &&   创建新类
SET CLASSLIB TO myclslib ADDITIVE   && 打开 myclsLib.VCX
RELEASE CLASSLIB myclslib
```

请参阅

ADD CLASS, CREATE CLASS, CREATE CLASSLIB, MODIFY CLASS, REMOVECLASS , RENAME CLASS, SET CLASSLIB

RELEASE LIBRARY 命令

从内存中删除一个外部 API 库。

语法

RELEASE LIBRARY LibraryName

参数描述

LibraryName

指定要从内存中删除的 API **库的名称。**

说明

发出 SET LIBRARY TO **命令，可以从内存中删除所有** API **库。用** SET LIBRARY **命令可以把某个外部的** API **例程库加载到内存。**

请参阅

SET LIBRARY

RELEASE MENUS **命令**

从内存中删除用户自定义菜单栏。

语法

RELEASE MENUS [MenuBarNameList [EXTENDED]]

参数描述

MenuBarNameList
指定要从内存中删除的各个菜单栏，用逗号分隔各菜单栏的名称。

EXTENDED
释放菜单栏及其下属的所有菜单、菜单标题、菜单项和全部相关的 ON SELECTION BAR，ON SELECTION MENU，ON SELECTION PAD，及 ON SELECTION POPUP 命令。

说明

要从内存中释放一个活动的菜单栏，必须首先用 DEACTIVATE MENU 命令使该菜单栏不活动。

如果发出不带任何附加参数的 RELEASE MENUS 命令，则从内存中删除所有的用户自定义菜单栏。

请参阅

DEFINE MENU

Release 方法

从内存中释放表单集或表单。

语法

Object.Release

说明

当用 DO FORM 命令创建表单集或表单，并且不存在可引用该表单集或表单的变量时，Release 方法很有效。可以使用 Screen 对象的 Forms 集合找到表单集或表单，并调用其 Release 方法。

可以使用 Screen 对象的表单集合来查找表单或表单集，并且调用 Release 方法。

应用于

表单，表单集，_SCREEN

请参阅

DO FORM

RELEASE PAD 命令

从内存中删除指定或所有的菜单标题。

语法

RELEASE PAD MenuTitleName OF MenuBarName | ALL OF MenuBarName

参数描述

MenuTitleName OF MenuBarName

指定要从内存中删除的菜单标题。

可以在 *MenuTitleName* 中指定 Visual FoxPro 系统菜单栏上的某个菜单标题而删除该菜单标题。例如，微软系统菜单的 RELEASE PAD _MEDIT 命令从 Visual FoxPro 系统菜单栏删除了编辑菜单标题。

ALL OF MenuBarName

指定从内存中删除用户自定义菜单栏上的每个菜单标题。ALL 子句不能用来从 Visual FoxPro 系统菜单栏上删除菜单标题。

示例

下面的 example 从系统菜单栏上删除“窗口”菜单标题。

```
PUSH MENU _MSYSMENU
RELEASE PAD _MSM_WINDO OF _MSYSMENU && 删除“窗口”菜单标题

WAIT WINDOW '按键恢复默认菜单'
POP MENU _MSYSMENU && 还原默认的 Visual FoxPro 菜单系统
```

请参阅

DEFINE PAD, RELEASE BAR, System Menu Names

RELEASE POPUPS 命令

从内存中删除指定或所有的菜单。

语法

RELEASE POPUPS [MenuNameList [EXTENDED]]

参数描述

MenuNameList

指定要从内存中释放的菜单，菜单名之间用逗号分隔。

也可以释放 Visual FoxPro 系统菜单栏上的 Visual FoxPro 系统菜单。要释放 Visual FoxPro 系统菜单，应包含系统菜单的内部名称(_MFILE，_MEDIT，_MDATA 等等）。可以用 SET SYSMENU TO DEFAULT 命令还原默认的系统菜单栏及系统菜单。

EXTENDED

释放菜单及其菜单项和所有与 ON SELECTION POPUP 及 ON SELECTION BAR 相关的命令。

说明

要从内存中释放一个活动菜单，必须先用 DEACTIVATE POPUP 命令使该菜单不活动。

如果 RELEASE POPUPS 命令没有其他的附加参数描述，则从内存中删除所有的用户

自定义菜单。

请参阅

DEFINE POPUP, System Menu Names

RELEASE PROCEDURE 命令

关闭用 SET PROCEDURE 命令打开的过程文件。

语法

RELEASE PROCEDURE FileName1 [, FileName2 ...]

参数描述

FileName1 [, FileName2 ...]

指定要关闭的过程文件名称。

说明

过程文件可以用 SET PROCEDURE 命令打开。一旦打开，过程文件中的过程即可在程序和命令窗口中使用。要关闭所有打开的过程文件，应发出不带任何附加参数的 SET PROCEDURE TO 命令。

发出不带任何的附加参数描述的 SET PROCEDURE TO 命令可以关闭所有打开的过程

文件。

请参阅

SET PROCEDURE

RELEASE WINDOWS 命令

从内存中释放用户自定义窗口或 Visual FoxPro 主窗口。

语法

RELEASE WINDOWS [WindowNameList]

参数描述

WindowNameList

指定要从内存中释放的窗口。*WindowNameList* 可以同时包含用户自定义窗口和 Visual FoxPro 主窗口。如果 *WindowNameList* 包括多个窗口，应用逗号分隔窗口名称。

如果没有包含 *WindowNameList*，则释放当前活动的用户自定义窗口。

RELEASE WINDOWS 可以用来从 Visual FoxPro 主窗口、父窗口或用户自定义窗口中释放 Visual FoxPro 系统窗口。

下表列出了可以从 Visual FoxPro 主窗口或父窗口释放的系统窗口：

- Command
- Debug
- Trace
- View

释放系统窗口与/或工具栏(在 Visual FoxPro **中)，在系统窗口或工具栏名称加上引号。例如，下列命令可以释放** Visual FoxPro **的报表控件工具栏：**

```
RELEASE WINDOW "Report Controls"
```

用 ACTIVATE WINDOW **在** Visual FoxPro **主窗口或用户自定义窗口中放置一个系统窗口。**

请参阅

ACTIVATE WINDOW, DEFINE WINDOW

ReleaseType **属性**

返回一个整数，以此确定表单对象如何释放。设计时不可用，运行时只读。

语法

Object.ReleaseType

参数描述

ReleaseType 属性的设置有：

设置	说明
0	变量被释放。
1	关闭框。
2	退出 Visual FoxPro。

说明

ReleaseType 属性在调用 QueryUnload 事件之前被设置。

应用于

表单，_SCREEN

请参阅

QueryUnload 事件，Unload 事件

返回总目录

Remove 方法

从文件集合和项目中移去一个文件。

语法

Object.Remove( )

说明

如果文件成功地从项目中移去了，则 Remove 方法返回“真” (.T.)；否则返回“假” (.F.)。注意，对于包含源代码管理器中的文件的项目，当您使用 Remove 方法从该项目中移去文件时，不会显示一个警告。

在从文件集合和项目中移去一个文件之前，会发生 QueryRemoveFile 事件。如果 QueryRemoveFile 事件返回“假” (.F.)，则不能从文件集合和项目中移去该文件，并且 Remove 方法返回“假” (.F.)。如果 QueryRemoveFile 方法返回“真” (.T.)，则从文件集合和项目中移去该文件，并且 Remove 方法返回“真” (.T.)。

应用于

文件对象

请参阅

Add 方法, Modify 方法, QueryRemoveFile 事件

RemoveFromSCC 方法

从源代码管理器中移去项目的一个文件。

语法

Object.RemoveFromSCC( )

说明

如果该文件成功地从源代码管理器中移去了，则 RemoveFromSCC 方法返回“真” (.T.)；否则返回“假” (.F.)。

应用于

文件对象

请参阅

AddToSCC 方法, CheckIn 方法, CheckOut 方法, GetLatestVersion 方法, UndoCheckOut 方法

REMOVE CLASS 命令

从 .VCX 可视类库中删除一个类定义。

语法

REMOVE CLASS ClassName OF ClassLibraryName

参数描述

ClassName

指定要从可视类库中删除的类定义名称。

OF ClassLibraryName

指定包含类定义的 .VCX 可视类库的名称。如果 *ClassLibraryName* 中没有包含文件扩展名，则假定其扩展名为 .VCX。

说明

删除类定义时请特别小心，因为要删除的类定义可能是用来派生其他类的父类。

请参阅

ADD CLASS, CREATE CLASS, CREATE CLASSLIB, MODIFY CLASS, RELEASE CLASSLIB, RENAME CLASS, SET CLASSLIB

REMOVE TABLE 命令

从当前数据库中移去一个表。

语法

REMOVE TABLE TableName | ?
[DELETE] [RECYCLE]

参数描述

TableName

指定要从当前数据库中移去的表。

?

显示“移去”对话框，从中可以选择一个要从当前数据库中移去的表。

DELETE

指定从数据库中移去该表，并从磁盘上删除。

警告 不能找回用此子句从磁盘中删除的表。即使 SETSAFETY 为 ON，在从磁盘上删除表时也不会有警告。

RECYCLE

指定不会立即从磁盘上删除表，而是放在 Microsoft Windows 回收站中。

说明

一个表从数据库中移去之后，便变成了自由表，因而可以添加到另一个数据库中。将

表添加到数据库中的命令是 ADD TABLE 。

REMOVE TABLE 命令要求以独占方式使用数据库。要以独占方式打开数据库，应在 OPEN DATABASE 命令中包含 EXCLUSIVE 子句。当执行了 REMOVE TABLE 命令后，所有与被索引的表有关的主索引、默认值和有效性规则也被删除。因此，在用此命令移去一个表时，如果 SET SAFETY 设置为 ON ，Visual FoxPro 会询问是否真的想从数据库中删除该表。

重要提示 如果 REMOVE TABLE 命令中删除的表与其他表存在相关的规则或关系，则此命令也会影响当前数据库中的这些表。从数据库中移去表后，这些相关的规则或关系不再有效。

示例

下面的示例创建了两个数据库，mydbc1 和 mydbc2，还有表 table1。表创建时被添加到 mydbc1。然后关闭表并且从 mydbc1 中删除它。再用 ADD TABLE 添加表到 mydbc2 中。 用 RENAME TABLE 将表名称从 table1 更改为 table2。

```
CREATE DATABASE mydbc1
CREATE DATABASE mydbc2
SET DATABASE TO mydbc1
CREATE TABLE table1 (cField1 C(10), n N(10))  && 向 mydbc1 数据库添加表
CLOSE TABLES       && 表经关闭后才能移到另一个数据库
REMOVE TABLE table1
SET DATABASE TO mydbc2
ADD TABLE table1
RENAME TABLE table1 TO table2
```

请参阅

ADD TABLE, CLOSE DATABASES, CREATE DATABASE, FREE TABLE, OPEN DATABASE

RemoveItem 方法

从组合框或列表框中移去一项。

语法

Control.RemoveItem(nIndex)

参数描述

nIndex

指定一个整数，它对应于被移去项在控件中的显示顺序。对于列表框或组合框中的第一项， *nIndex* = 1。

应用于

组合框，列表框

请参阅

AddItem **方法**, Clear **方法**, RemoveListItem **方法**

RemoveListItem 方法

从组合框或列表框中移去一项。

语法

Control.RemoveListItem(nItemId)

参数描述

nItemId

指定一个整数，它表示删除项在控件中的唯一标识。

应用于

组合框，列表框

请参阅

AddItem 方法, Clear 方法

RemoveObject 方法

运行时从容器对象中删除一个指定的对象。

语法

Object.RemoveObject(cObjectName)

参数描述

cObjectName

指定要删除的对象名，如果指定对象不存在，则会出错。

说明

对象删除后，它便从屏幕上消失，并且不能再引用。

应用于

列，命令组，容器对象，自定义，数据环境，表单，表单集，表格，选项按钮，页面，页框，_SCREEN**，工具栏**

请参阅

AddObject **方法**，RemoveItem **方法**，RemoveListItem **方法**

RENAME **命令**

更改文件名。

语法

RENAME FileName1 TO FileName2

参数描述

FileName1 TO FileName2

指定要重命名的文件名和新文件名。注意文件名中要包括扩展名。如果文件名中不包括扩展名，则假定默认的扩展名为 .DBF。如果重命名的表是具有 .FPT 备注文件的自由表，应同时重命名备注文件。当要重命名的文件确实没有扩展名时，应在文件名后加入句点 (.)。

不要使用 RENAME 命令重命名数据库中的表。RENAME 不能用来重命名数据库中的表。要重命名数据库中的表，可使用 RENAME TABLE。

如果要重命名的文件不在默认驱动器和目录中，请把路径包括在文件名中。如果 *FileName1* 和 *FileName2* 在不同目录中，此命令将把 *FileName1* 移到 *FileName2* 所在的目录中。

执行 RENAME 时，*FileName2* 不能是现有文件，而 *FileName1* 则必须存在并且没有打开。

FileName1 和 FileName2 可以包含通配字符，如 * 和 ?。例如，要重命名当前目录或文件夹中带 .prg 扩展名的程序文件为带 .bak 扩展名的备份文件，可用 RENAME *.prg TO *.bak 。

请参阅

COPY FILE, COPY TO, RENAME CLASS, RENAME TABLE

RENAME CLASS 命令

重命名 .VCX 可视类库中的一个类定义。

语法

RENAME CLASS ClassName1 OF ClassLibraryName TO ClassName2

参数描述

ClassName1

指定要重命名的类定义名称。

OF ClassLibraryName

指定 .VCX 可视类库的名称，其中包含了要重命名的类。如果在 *ClassLibraryName* 中没有指定文件扩展名，Visual FoxPro 自动指定扩展名为 .VCX。

TO ClassName2

指定类定义的新名称。

说明

重命名类定义时请小心，重命名类定义后，Visual FoxPro 并不更新可视类库中引用此类定义的其他类定义。

示例

下面的 example 用 CREATE CLASSLIB 命令创建了一个名为 myclslib 的可视类库。

然后用 CREATE CLASS 命令创建一个派生于 Visual FoxPro 基类、名为 myform 的类，并将它保存到 myclslib 可视类库中。最后用 RENAME CLASS 命令将此类的名称由 myform 更改为 yourform。

```
CREATE CLASSLIB myclslib      && 创建一个新的 .VCX 可视类库
CREATE CLASS myform OF myclslib AS "Form"   && 创建新类
RENAME CLASS myform OF myclslib TO yourform   && 更改类名
```

请参阅

ADD CLASS, CREATE CLASS, MODIFY CLASS, RELEASE CLASSLIB, REMOVE CLASS, SET CLASSLIB

RENAME CONNECTION 命令

重命名当前数据库中的一个命名连接。

语法

RENAME CONNECTION ConnectionName1 TO ConnectionName2

参数描述

ConnectionName1

指定要重命名的连接名称。

ConnectionName2

指定连接的新名称。

说明

重命名连接时，必须以独占方式打开包含此命名连接的数据库，并必须是当前数据库。要以独占方式打开数据库，可在 OPEN DATABASE 命令中包括 EXCLUSIVE 子句。

示例

下面的示例假设有名称为 MyFoxSQLNT 的 ODBC 数据源，并且此数据源的用户 ID 是“sa.”。打开数据库 testdata 并创建连接 Myconn1。DISPLAY CONNECTIONS 用于显示数据库中的命名连接。

再用 RENAME CONNECTION 更改新建的连接名为 Myconn2。DISPLAY CONNECTIONS 重新显示新名称的连接。DELETE CONNECTION 从数据库中删除重新命名的连接。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
CREATE CONNECTION Myconn1 DATASOURCE "MyFoxSQLNT" USERID "sa"

CLEAR
DISPLAY CONNECTIONS  && 显示数据库间已命名的联系
RENAME CONNECTION Myconn1 TO Myconn2
DISPLAY CONNECTIONS  && 显示新命名的联系

DELETE CONNECTION Myconn2  && 删除新命名的联系
```

请参阅

DELETE CONNECTION, DBGETPROP(), DISPLAY CONNECTIONS, LISTCONNECTIONS, MODIFY CONNECTION, OPEN DATABASE

RENAME TABLE 命令

重命名当前数据库中的表。

语法

RENAME TABLE TableName1 TO TableName2

参数描述

TableName1

指定要重命名的表名称。

TableName2

指定表的新名称。

说明

RENAME TABLE 不能用来更改自由表的名称。要更改自由表的名称，可使用 RENAME 命令。

示例

下面的示例创建了两个数据库，mydbc1 和 mydbc2，还有表 table1。表创建时被添加到 mydbc1。然后关闭表并且从 mydbc1 中删除它。再用 ADD TABLE 添加表到 mydbc2 中。用 RENAME TABLE 将表名称从 table1 更改为 table2。

```
CLOSE DATABASES
  CREATE DATABASE mydbc1
  CREATE DATABASE mydbc2
```

```
SET DATABASE TO mydbc1
CREATE TABLE table1 (cField1 C(10), n N(10))   && 向 mydbc1 数据库添加表
CLOSE TABLES        && 表经关闭后才能移动
REMOVE TABLE table1
SET DATABASE TO mydbc2
ADD TABLE table1
RENAME TABLE table1 TO table2
```

请参阅

ADD TABLE, CLOSE DATABASES, CREATE DATABASE, COPY FILE, COPY TO, OPEN DATABASE, RENAME

RENAME VIEW 命令

重命名当前数据库中的 SQL 视图。

语法

RENAME VIEW ViewName1 TO ViewName2

参数描述

ViewName1

指定要重命名的 SQL 视图名称。

ViewName2

指定 SQL 视图的新名称。

说明

重命名 SQL 视图之前，必须以独占方式打开包含 SQL 视图的数据库，并将该数据库设置为当前数据库。要以独占方式打开数据库，应在使用 OPEN DATABASE 时包括 EXCLUSIVE 子句。

示例

下面的示例打开了数据库 testdata。用 CREATE SQL VIEW 创建一个本地的 SQL 视图 myview，它是从选择了 customer 表中的所有记录的 SELECT – SQL 语句创建的。RENAME VIEW 重命名视图的名称 myview 为 yourview。显示“视图设计器”，允许修改新重命名的 SQL 视图。关闭“视图设计器”后，SQL 视图被删除。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')

CREATE SQL VIEW myview AS SELECT * FROM customer   && 创建一个新视图
RENAME VIEW myview TO yourview   && 改变视图名称
MODIFY VIEW yourview   && 打开视图设计器
DELETE VIEW yourview   && 删除视图
```

请参阅

CREATE SQL VIEW, DELETE VIEW, DISPLAY VIEWS, LIST VIEWS, MODIFY VIEW, OPEN DATABASE, RENAME VIEW, SELECT – SQL, USE

REPLACE 命令

更新表的记录内容。

语法

```
REPLACE FieldName1 WITH eExpression1 [ADDITIVE]
    [, FieldName2 WITH eExpression2 [ADDITIVE]] ...
    [Scope] [FOR lExpression1] [WHILE lExpression2]
    [IN nWorkArea | cTableAlias]
    [NOOPTIMIZE]
```

参数描述

FieldName1 WITH eExpression1 [, FieldName2 WITH eExpression2 ...]

指定用表达式 *eExpression1* 的值来代替 *FieldName1* 字段中的数据；用表达式 *eExpression2* 的值来代替字段 *FieldName2* 中的数据，依此类推。

当表达式的值比数值字段的宽度长时，REPLACE 采用以下方法来处理数据：

1. 首先，REPLACE 截短表达式的小数位然后圆整剩余部分。
2. 如果此时字段仍然放不下表达式的值，则 REPLACE 用科学计数法在字段中保存表达式的值。
3. 如果还不行，REPLACE 则用星号代替字段内容。

ADDITIVE

把对备注字段的替代内容追加到备注字段的后面。ADDITIVE 只对替换备注字段有用。如果省略 ADDITIVE，则用表达式的值改写备注字段原有内容。

Scope

指定要替换内容的记录范围。只替换指定范围内记录字段的内容。Scope 子句有：ALL、NEXT *nRecords*、RECORD *nRecordNumber* 和 REST。有关 scope 子句的详细内容，请参阅帮助中的“Scope Clauses”和“语言概述”。

REPLACE 的默认范围是当前记录 (NEXT 1)。

FOR lExpression1

只有当指定记录使表达式 *lExpression1* 求值结果为“真”(.T.) 时，它的字段才会被替换为新的内容。因此，包含 FOR 子句可以使命令有条件地更新记录，而将那些不需要更新的记录筛选掉。

当表达式 *lExpression1* 是可优化表达式时，Rushmore 将优化 REPLACE 命令。因此，为使系统获得最佳性能，应在 FOR 表达式中使用可优化的表达式。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

WHILE lExpression2

指定一个逻辑表达式 *lExpression2* 作为替换字段内容的条件。只要逻辑表达式 *lExpression2* 计算为“真”(.T.)，就替换记录中的数据，直至遇到第一个计算结果为“假”(.F.) 的表达式为止。

IN nWorkArea

指定要更新记录的表所在的工作区。

IN cTableAlias

指定要更新记录的表的别名。

如果同时省略 *nWorkArea* 和 *cTableAlias*，则更新当前选定工作区中表的记录。

NOOPTIMIZE

关闭 Rushmore 优化。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

说明

REPLACE 命令用表达式的值替换字段中的数据。在替换未选定工作区中的字段时，字段前面必须加上表的别名。

注意 如果记录指针已在当前工作区中文件的末端，而指定的字段在另一个工作区中，则不发生任何替换。

示例

下面的示例包含 10 个记录的表。用 REPLACE 在字段中放置随机值。MIN( ) 和 MAX( ) 用来显示表中的最大值和最小值。

```
CLOSE DATABASES
CREATE TABLE Random (cValue N(3))
FOR nItem = 1 TO 10   && 添加 10 条记录
   APPEND BLANK
   REPLACE cValue WITH 1 + 100 * RAND( )   && 插入任意值
ENDFOR

CLEAR
LIST   && 显示这些值
gnMaximum = 1   && 初始化最小值
```

```
gnMinimum = 100   && 初始化最大值
SCAN
    gnMinimum = MIN(gnMinimum, cValue)
    gnMaximum = MAX(gnMaximum, cValue)
ENDSCAN
? 'The minimum value is: ', gnMinimum   && 显示最小值
? 'The maximum value is: ', gnMaximum   && 显示最大值
```

请参阅

GATHER, INSERT-SQL, SCATTER

REPLACE FROM ARRAY 命令

使用变量数组中的值更新字段内容。

语法

```
REPLACE FROM ARRAY ArrayName
    [FIELDS FieldList]
    [Scope]
    [FOR lExpression1]
    [WHILE lExpression2]
  [NOOPTIMIZE]
```

参数描述

ArrayName

指定用来替换字段内容的数组名称。

FIELDS FieldList

指定只有 *FieldList* 中的字段才被数组中的内容替换。当替换非选定工作区的字段时，必须在字段前加上所在表的别名。

Scope

指定一个范围，规定要用数组中的内容替换其字段内容的记录。只有当记录在指定范围内时，才用数组内容替换它的字段值。替换一直进行到范围的末端或数组的末端。

Scope 子句有：ALL，NEXT *nRecords*，RECORD *nRecordNumber* 和 REST。有关 scope 子句的详细内容，请参阅帮助中的“Scope Clauses”和“语言概述”。

REPLACE FROM ARRAY 的默认范围是当前记录 (NEXT 1)。

FOR lExpression1

指定只有记录使表达式 *lExpression1* 计算值为“真”(.T.)时，才替换记录中字段的内容。包含 FOR 子句是为了有条件地替换记录，筛选出不想替换的记录。对于每一个记录，只要它使表达式 *lExpression1* 的值计算为“真”(.T.)，且数组中有元素，替换就要进行。

如果 *lExpression1* 是一个可优化表达式，Rushmore 将优化 REPLACE FROM ARRAY 命令。为获得最佳运行性能，应在 FOR 表达式中使用可优化表达式。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版

程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

WHILE lExpression2

指定一个逻辑表达式 *lExpression2* 作为替换字段内容的条件。只要 *lExpression2* 计算为“真”，就替换记录中字段的数据，直至遇到使表达式不为“真”(.T.) 的记录为止。

NOOPTIMIZE

关闭 Rushmore 优化。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

说明

REPLACE FROM ARRAY 命令不处理备注和通用字段。若要往这些字段中加入数据，应使用 GATHER 和 APPEND GENERAL 命令。

替换从数组中的第一个元素开始，用数组中的元素替换记录中相应字段的内容。数组的第一个元素替换记录的第一个字段，第二个元素替换记录的第二个字段，依此类推。如果数组的元素比表的字段少，则不替换多余的字段。如果数组的元素比表的字段多，则忽略数组中多余的元素。

注意 如果记录指针在当前工作区中文件的末端，而指定的字段在另一个工作区，则不发生任何替换。

请参阅

APPEND FROM ARRAY, APPEND GENERAL, COPY TO ARRAY, GATHER, REPLACE, SCATTER, SET OPTIMIZE

REPLICATE() 函数

返回一个字符串，这个字符串是将指定字符表达式重复指定次数后得到的。

语法

REPLICATE(cExpression, nTimes)

返值类型

字符型

参数描述

cExpression

指定要重复的字符表达式。

nTimes

指定字符表达式的重复次数。

说明

在 Visual FoxPro 中，结果字符串的最大长度只受可用内存数量的限制。

示例

```
CLEAR
? REPLICATE('HELLO ',4) && 显示 HELLO HELLO HELLO HELLO
```

请参阅

SPACE()

REPORT 命令

根据 MODIFY REPORT **或** CREATE REPORT **创建的报表定义文件显示或打印报表。**

语法

```
REPORT FORM FileName1 | ?
    [ENVIRONMENT]
    [Scope] [FOR lExpression1] [WHILE lExpression2]
    [HEADING cHeadingText]
    [NOCONSOLE]
    [NOOPTIMIZE]
    [PLAIN]
    [RANGE nStartPage [, nEndPage]]
    [PREVIEW [[IN] WINDOW WindowName | IN SCREEN]
    [NOWAIT]]
    [TO PRINTER [PROMPT] | TO FILE FileName2 [ASCII]]
    [NAME ObjectName]
  [SUMMARY]
```

参数描述

FileName1

指定报表定义文件的名称。

?

显示“打开”对话框，从中可选择报表文件。

ENVIRONMENT

包括此子句是为了提供与 2.x 版本报表的向后兼容性。要恢复与 Visual FoxPro 报表相关的数据环境，应把数据环境 AutoOpenTables 属性设置为默认值“真”(.T.)。若要确保在报表打印完后关闭报表环境，应把数据环境 AutoCloseTables 属性设置为默认值“真”(.T.)。

对于从早期 FoxPro 版本转换过来的报表，即便 AutoOpenTables 属性设置为“假”(.F.)，包含 ENVIRONMENT 也会打开并恢复数据环境中的所有表和关系。

当创建或修改报表时，可以用报表定义文件保存当前 Visual FoxPro 数据环境。保存 Visual FoxPro 数据环境，实际上是在报表定义表中添加了一些附加的记录，保存下列信息：所有打开的表和索引文件、索引顺序以及表之间关系。

Scope

指定要包含在报表中的记录范围。只有在指定范围内的记录才包括在报表中。

Scope 子句有：ALL、NEXT *nRecords*、RECORD *nRecordNumber* 和 REST。有关 scope 子句的详细内容，请参阅帮助中的“Scope Clauses”和“语言概述”。

REPORT 的默认范围是所有记录(ALL)。

FOR lExpression1

只有使表达式 *lExpression1* 的计算值为“真”(.T.)的记录，才打印其中的数据。包括 FOR 可以筛选出不想打印的记录。

如果 *lExpression1* 是一个可优化表达式，Rushmore 将优化 REPORT FOR 命令。为获得最佳运行性能，应在 FOR 表达式中使用可优化表达式。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

WHILE lExpression2

指定一个逻辑表达式 *lExpression2* 作为打印数据的条件。只要 *lExpression2* 条件计算为“真”(.T.)，就打印记录中的数据，直至遇到使表达式不为“真”(.T.)的记录为止。

HEADING cHeadingText

指定放在报表每页上的附加标题文本。如果既包括 HEADING 又包括了 PLAIN，应把 PLAIN 子句放在前面。

NOCONSOLE

当打印报表或将报表传输到一个文件时，不在 Visual FoxPro 主窗口或用户自定义窗口中显示有关信息。

NOOPTIMIZE

若要关闭对 REPORT 命令的 Rushmore 优化，应包括 NOOPTIMIZE 子句。

详细内容，请参阅稍后的 SET OPTIMIZE 命令与《Microsoft Visual FoxPro 6.0 中文版程序员指南》第十五章“优化应用程序”中的“掌握 Rushmore 技术”。

PLAIN
指定只在报表开始位置出现的页标题。
RANGE nStartPage [, nEndPage]
指定要打印的页码范围。nStartPage 指定了要打印的第一页；nEndPage 指定了要打印的最后一页。如果省略 nEndPage，则要打印的最后一页默认为 9,999。
PREVIEW [[IN] WINDOW WindowName | IN SCREEN]
以页面预览模式显示报表，而不把报表送到打印机中打印。要打印报表，必须发出带 TO PRINTER 子句的 REPORT 命令。
请注意：当命令中包括 PREVIEW 子句时，忽略系统变量。
使用可选的 WINDOW 或 IN WINDOW 子句中，您可以指定一个窗口 WindowName，报表输出到这个窗口中。该窗口可由 DEFINE WINDOW 命令定义。如果包含 WINDOW 子句，则使用 WindowName *指定的*窗口的特性（例如标题、大小等等）进行预览。如果包含了 IN WINDOW 子句，则在 WindowName *指定的*窗口中预览报表。
包含可选的 IN SCREEN 子句，表明报表预览窗口位于 Visual FoxPro 主窗口中，并且不能移动到外面去。
可以在命令中包括可选的 NOWAIT 子句。这时，Visual FoxPro 能够在运行程序时不等待关闭页面预览窗口就继续执行程序。也就是说，当页面窗口打开时，Visual FoxPro 继续运行程序。
在已发布的应用程序中，应确保可以得到“查看”菜单。关掉打印预览工具栏时，如果没有“查看”菜单，将无法恢复“打印预览”工具栏。

TO PRINTER [PROMPT]

把报表输送到打印机打印。

在 Visual FoxPro 中，命令中可以包括可选的 PROMPT 子句，在打印开始前显示设置打印机的对话框。可调整的打印设置项取决于当前安装的打印机驱动程序。PROMPT 子句应紧跟在 TO PRINTER 子句之后。

TO FILE FileName2 [ASCII]

在 Visual FoxPro 中，指定报表要送往的文本文件。将报表送往文本文件时，使用当前打印机驱动程序。包含 TO FILE 子句创建的文件具有默认扩展名 .TXT。

在 Visual FoxPro 中，可以在命令中包括可选 ASCII 子句，用报表定义文件创建一个 ASCII 文本文件。没有 ASCII 子句时，则按 PostScript® 或其他打印机代码格式将报表写到文本文件中。报表定义中任何图像、线条、矩形以及圆角矩形都不出现在 ASCII 文件中。

在 ASCII 文本文件的每一页中，列和行的数目由系统变量 _ASCIICOLS 和 _ASCIIROWS 的内容确定。_ASCIICOLS 和 _ASCIIROWS 的默认值分别为 80 列和 63 行。这些值对应于标准的纵向页。

NAME ObjectName

给报表的数据环境指定一个对象变量名。数据环境和数据环境中的对象都有一些属性和方法，例如 AddObject ，这些属性和方法可在运行时设置或调用。对象变量则提供了存取这些属性和方法的途径。如果不指定 NAME 子句，Visual FoxPro 则使用报表文件的名称作为默认对象变量名，此默认名也可以在事件代码中引用。

SUMMARY
不打印细节行，只打印总计和分类总计信息。

说明

报表定义文件的默认扩展名是 .FRX。如果报表定义文件不在默认目录中，文件名中必须包括文件的路径。

请参阅

_ASCIICOLS, _ASCIIROWS, CREATE REPORT, DataEnvironment 对象, MODIFY REPORT

Requery 方法

重新查询列表框或组合框控件所基于的行源 (row source)。

语法

Control.Requery

说明

使用 Requery 方法可以确保控件中包含最新的数据。Requery 方法重新查询 RowSource 属性，并且使用新的值更新列表。

应用于

组合框，列表框

请参阅

RowSource **属性**

REQUERY() **函数**

为远程 SQL **视图再次检索数据。**

语法

REQUERY([nWorkArea | cTableAlias])

参数描述

nWorkArea

指定打开远程 SQL **视图的工作区。**

cTableAlias

指定远程 SQL **视图的别名。如果省略了** *nWorkArea* **和** *cTableAlias*，**则检索在当前选定工作区中打开的** SQL **视图的数据。**

返值类型

数值型

说明

如果检索数据成功，REQUERY() **返回** 1，**否则，返回** 0。

REQUERY **通常用来在数据源发生变化时刷新** SQL **视图。**

请参阅

CREATE SQL VIEW, SELECT-SQL, USE

RequestData 方法

创建一个数组，其中包含来自在 Visual FoxPro 的一个实例中打开表的数据。

语法

ApplicationObject.RequestData([nWorkArea | cTableAlias] [, nRecords])

返值类型

Array

参数描述

nWorkArea

指定表的工作区编号，将该表的数据保存在数组中。如果省略 cTableAlias 和 nWorkArea，则将在当前工作区中打开的表的数据保存在数组中。

cTableAlias

指定表的别名，将该表的数据保存在数组中。

nRecords

指定在数组中保存的记录数，从当前记录开始。如果省略 nRecords，并且有足够的内存，则从当前记录开始，将所有的记录保存在数组中。

说明

使用 RequestData 方法可以从 Visual FoxPro 的一个实例获取数据。

示例

下面的示例从 Visual FoxPro 中运行，创建了 Visual FoxPro 的第二个实例。在 Visual FoxPro 的第二个实例中打开 Customer 表。

在 Visual FoxPro 的第一个实例中创建从 Customer 表获得数据的数组，并且显示该数组的内容。此数组包含从 Customer 表的前五个记录获得的数据。然后关闭 Visual FoxPro 的第二个实例。

```
oNewInstance = CREATEOBJECT('VisualFoxPro.Application')
oNewInstance.DoCmd "USE (HOME(2) + 'Data\Customer')"
aCustomerArray = oNewInstance.RequestData('Customer',5)
DISPLAY MEMORY LIKE aCustomerArray
oNewInstance.DoCmd('QUIT')
```

应用于

Application 对象，_VFP 系统变量

请参阅

DataToClip 方法

Reset 方法

重置计时器控件，让它从 0 开始。

语法

Timer.Reset

应用于

计时器

请参阅

Interval 属性, Timer 事件

ResetToDefault 方法

将属性、事件或方法恢复到 Visual FoxPro 默认设置。运行和设计时可用。

语法

[Form.]Object.ResetToDefault(cPropertyName
| cEventName | cMethodName)

参数描述

cPropertyName

指定要恢复为 Visual FoxPro 默认设置的属性的名称。

cEventName

指定要恢复的事件的名称。将删除该事件中的所有用户自定义代码。

cMethodName

指定要恢复的方法的名称。将删除该方法中的所有用户自定义代码。

说明

ResetToDefault 将一个属性重新设置为创建时的值。例如，如果更改了一个命令按钮的字体，则调用标题的该方法重置字体为默认值 (Arial)。

ResetToDefault 在设计时删除事件或方法中的所有用户自定义代码。

应用于

ActiveDoc，复选框，列，组合框，命令按钮，命令按钮组，容器对象，控件对象，临时表，自定义对象，数据环境，编辑框，表单，表单集，表格，标头，图像，标签，线条，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，选项按钮组，页面，页框，项目挂接对象，关系，屏幕，形状，微调，文本框，计时器，工具栏

请参阅

GETPEM()

Resizable 属性

指定列对象的大小能否在运行时由用户调节。设计时可用，运行时可读写。

语法

Column.Resizable[= lExpr]

参数描述

lExpr

Resizable 属性的设置有：

设置	说明
“真” (.T.)	列对象可改变大小。
“假” (.F.)	（默认值）列对象的大小不能由用户改变。

应用于

列

请参阅

Width 属性

Resize 事件

当调整对象大小时发生。

语法

PROCEDURE Object.Resize
[LPARAMETERS nIndex]

参数描述

nIndex

唯一标识控制数组中的控制。

说明

Resize 事件可以由交互式方式触发，也可以在代码中通过设置 Height 和 Width 属性来触发。对于列，Resize 事件可以通过设置列的 Width 属性来触发。

应用于

列，容器对象，控制对象，表单，表格，OLE 绑定型控制，OLE 容器控制，页框，工具栏

请参阅

Height 属性, Paint 事件, Width 属性

RESTORE FROM 命令

恢复保存在变量文件或备注字段中的变量和变量数组。

语法

RESTORE FROM FileName | MEMO MemoFieldName
[ADDITIVE]

参数描述

FileName

指定保存变量的变量文件。变量文件具有扩展名 .MEM。

MEMO MemoFieldName

指定保存变量和变量数组的备注字段。

ADDITIVE

防止删除当前内存中已有的变量或变量数组。如果使用 ADDITIVE 时，如果要添加的变量或变量数组的数目加上已有变量的数目超过了系统对变量数目的限制，Visual FoxPro 将从变量文件或备注字段中恢复尽可能多的变量和变量数组。

恢复变量或变量数组时，如果变量或变量数组与已有变量或变量数组有相同的名称，则用恢复的变量或变量数组的值改写原有变量或变量数组中的值。

说明

当在程序中执行 RESTORE FROM 命令时，所有 PUBLIC 和 PRIVATE 变量和数组都恢复成 PRIVATE 变量或 PRIVATE 数组，所有 LOCAL 变量和数组都恢复成 LOCAL 变量或 LOCAL 数组。如果在命令窗口中执行 RESTORE 命令，则所有 PUBLIC 和 PRIVATE 变量和数组都恢复成 PUBLIC 变量或数组，所有 LOCAL 变量和数组都恢复成 LOCAL 变量或数组。

如果命令中没有包括 ADDITIVE 关键字，RESTORE FROM 将清除当前内存中的所有变量或数组。RESTORE FROM 命令不影响系统变量。

注意 从变量文件或备注字段中不能恢复这个对象类型变量。

示例

下面的示例中创建了两个变量。将它们保存到一个变量文件中，并且不清除现有的变量而恢复。

```
gnVal1 = 50
gcVal2 = 'Hello'
SAVE TO temp
CLEAR MEMORY

gdVal3 = DATE( )
RESTORE FROM temp ADDITIVE
CLEAR
DISPLAY MEMORY LIKE g*
```

请参阅

DIMENSION, PUBLIC, PRIVATE, RELEASE, SAVE TO, STORE

RESTORE MACROS 命令

把保存在键盘宏文件或备注字段中的键盘宏恢复到内存中。

语法

RESTORE MACROS
[FROM FileName | FROM MEMO MemoFieldName]

参数描述

FROM FileName

指定保存宏的宏文件。宏文件有 .FKY 扩展名，并且在 *FileName* 中没有指定扩展名时，用它作为文件的扩展名。如果要指定一个不同的宏文件扩展名，必须在 *FileName* 中包括扩展名。

FROM MEMO MemoFieldName

指定保存宏的备注字段。

说明

要把键盘宏保存到键盘宏文件或备注字段中，可使用 SAVE MACRO。

默认情况下，从文件或备注字段中恢复宏时采用追加方式，即把要恢复的宏追加到内存中已有宏的集合中。但如果宏文件或备注字段中的宏与内存中已有的宏同名，则用文件或备注字段中的宏改写内存中已有的宏。

如果执行 RESTORE MACROS 命令时没有文件名也没有备注字段名时，RESTORE

MACROS 命令将清除内存中所有的宏，并且恢复所有默认宏。默认的宏是从名为 DEFAULT.FKY 的键盘宏文件中恢复的。如果 Visual FoxPro 找不到 DEFAULT.FKY 文件，Visual FoxPro 将 F2 键到 F9 键之间的所有键恢复为标准 Visual FoxPro 启动设置。

请参阅

CLEAR, PLAY MACRO, SAVE MACROS

RESTORE SCREEN 命令

恢复原来保存在屏幕缓冲区、变量或数组元素中的 Visual FoxPro 主窗口或用户自定义窗口。

语法

RESTORE SCREEN
[FROM VarName]

参数描述

FROM VarName

指定变量或数组元素的名称，从中恢复屏幕或窗口图像。

说明

使用 SAVE SCREEN 命令，可以把当前的 Visual FoxPro 主窗口或活动的用户自定义窗

口保存到屏幕缓冲区、变量或数组元素中。

当使用 DISPLAY 或 LIST MEMORY 命令查看变量或数组元素时，保存屏幕或窗口图像的变量或数组元素具有 S 数据类型。也可以用 SAVE TO 命令把保存在变量或数组元素中的 Visual FoxPro 主窗口或用户自定义窗口保存到变量文件中，并且能用 RESTORE FROM 命令从变量文件中恢复它们。

如果执行 RESTORE SCREEN 命令时不带 FROM 子句，则 RESTORE SCREEN 从屏幕缓冲区中恢复 Visual FoxPro 主窗口或用户自定义窗口。

请参阅

RESTORE FROM, SAVE SCREEN, SAVE TO

RESTORE WINDOW 命令

将保存在窗口文件或备注字段中的窗口定义和窗口状态恢复到内存中。

语法

RESTORE WINDOW WindowNameList | ALL
FROM FileName | FROM MEMO MemoFieldName

参数描述

WindowNameList

指定一个或多个要恢复的窗口。多个窗口的名称之间用逗号隔开。

ALL

恢复窗口文件或备注字段中的所有窗口定义。

FROM FileName

指定要从中恢复窗口的窗口文件。窗口文件具有 .WIN 扩展名。如果保存窗口时给了窗口文件一个不同于 .WIN 的扩展名，则必须在 *FileName* 中加入扩展名。

FROM MEMO MemoFieldName

指定要从中恢复窗口的备注字段。

说明

使用 SAVE WINDOW 命令，可以把窗口定义保存在窗口文件或备注字段中。内存中任何与要恢复窗口同名的窗口都被改写。窗口在保存时的状态（隐藏的、活动的等等）在恢复时都被恢复出来。

示例

在下面的示例中，定义了一个窗口 wOutput1 并且保存到一个变量中。清除所有的窗口，还原和激活窗口 wOutput1。

```
CLEAR
DEFINE WINDOW wOutput1 FROM 2,1 TO 13,75 TITLE 'Output' ;
    CLOSE FLOAT GROW ZOOM
SAVE WINDOW wOutput1 TO temp
CLEAR WINDOWS
RESTORE WINDOW wOutput1 FROM temp
ACTIVATE WINDOW wOutput1
WAIT "The window wOutput1 has been restored" WINDOW
  RELEASE WINDOW wOutput1
```

请参阅

ACTIVATE WINDOW, DEFINE WINDOW, SAVE WINDOWS

RESUME 命令

继续执行一个挂起的程序。

语法

RESUME

说明

使用 SUSPEND **可以挂起程序。挂起的程序重新运行时，从原来挂起时正在执行的行开始。**

SUSPEND **和** RESUME **是非常有用的调试工具。程序在执行时，可以将它挂起，以便检查** Visual FoxPro **环境的当前状态。这些状态包括变量、窗口、菜单栏以及菜单定义等。**

重要提示 **为防止程序挂起时执行的命令干扰后面的程序输出，应在恢复程序运行前执行** CLEAR **命令，清除** Visual FoxPro **主窗口或活动的用户自定义窗口。**

请参阅

CANCEL, CLEAR, RETURN, SUSPEND

RETRY 命令

重新执行前面的命令。

语法

RETRY

说明

RETRY 将控制权返回给调用程序，并且重新执行调用程序中最后执行过的程序行。RETRY 与 RETURN 相似，所不同的是 RETURN 返回到调用程序后，接着执行的是调用程序的下一行，而不是最后执行过的程序行。

RETRY 最有用的地方是错误处理例程，在锁定记录或文件时，它常用来重复执行一条命令，直到记录或文件锁定函数能够成功为止。可以使用 SET REPROCESS 来控制重试记录或文件锁定函数。在大多数网络环境中，SET REPROCESS 命令很有用。

请参阅

ON ERROR, RETURN, SET REPROCESS

RETURN 命令

将程序控制权返回给调用程序。

语法

RETURN [eExpression | TO MASTER | TO ProcedureName]

参数描述

eExpression

指定返回给调用程序的表达式。如果省略 RETURN 命令或省略返回表达式，则自动将“真”(.T.) 返回给调用程序。

TO MASTER

将控制返回给最高层的调用程序。

TO ProcedureName

将控制权返回给指定过程。

说明

RETURN 终止程序、过程或函数的运行，并将控制返回给调用程序、最高次调用程序、另一个程序或命令窗口。

当执行 RETURN 命令时，Visual FoxPro 释放 PRIVATE 类型的变量。

通常，RETURN 放在程序、过程或函数的末尾，用来将控制返回给高层的程序。但是，如果省略 RETURN 命令，一个隐含的 RETURN 命令也将被执行

示例

在下面的示例中，函数 longdate **返回一个适合于从日期中打印的字符串。**

```
SET CENTURY ON
? longdate({^1998-02-16})   && 显示周一，1998 年 2 月 16 日

FUNCTION longdate
PARAMETER mdate
RETURN CDOW(mdate) + ', ' + MDY(mdate)
```

请参阅

DO, FUNCTION, LPARAMETERS, PARAMETERS, PARAMETERS(), PRIVATE, PROCEDURE, PUBLIC

RGB() 函数

根据一组红、绿、蓝颜色成份返回一个单一的颜色值。

语法

RGB(nRedValue, nGreenValue, nBlueValue)

参数描述

nRedValue

指定红色成份的深度。*nRedValue* 的大小范围是 0 到 255。0 是最小的颜色强度，255 是最大的颜色深度。

nGreenValue

指定绿色成份的深度。*nGreenValue* 的大小范围是 0 到 255。

nBlueValue

指定蓝色成份的深度。*nBlueValue* 的大小范围是 0 到 255。

说明

RGB( ) 函数返回的值可以用来设置诸如 BackColor 和 ForeColor 等颜色属性。

示例

下面的示例用 RGB( ) 将表单的背景颜色更改为蓝色。

```
goMyForm = CREATEOBJECT('FORM')  && 创建一个表单
goMyForm.Show  && 显示表单
WAIT WINDOW 'Press a key to change the form color'
goMyForm.BackColor=RGB(0,0,255)  && 改变表单背景颜色
WAIT WINDOW 'Press a key to release the form'
RELEASE goMyForm  && 从存储中释放表单
```

请参阅

GETCOLOR( ), RGBSCHEME( )

RGBSCHEME() 函数

返回指定配色方案中的 RGB 颜色对或 RGB 颜色对列表。

语法

RGBSCHEME(nColorSchemeNumber [, nColorPairPosition])

返值类型

字符型

参数描述

nColorSchemeNumber

指定欲得到其完整 RGB 颜色列表的配色方案编号。RGBSCHEME( ) 返回 10 个 RGB 颜色对。

nColorPairPosition

返回配色方案中的某一个 RGB 颜色对。*nColorPairPosition* 指定 RGB 颜色对在配色方案中的位置。例如，如果 *nColorPairPosition* 是 4，则返回第四个颜色对。

说明

使用 SCHEME( ) 函数可以返回配色方案中传统的颜色对或颜色对列表。在 RGB 颜色对中，使用数值指定颜色。而传统的颜色对则使用字母来指定颜色。

示例

下例显示第四个配色方案中的第三个颜色对。

```
CLEAR
? RGBSCHEME(4,3)
```

请参阅

颜色概览, RGB( ), SCHEME( )

RIGHT() 函数

从字符串中的最右边开始返回指定数目的字符。

语法

RIGHT(cExpression, nCharacters)

返值类型

字符型

参数描述

cExpression

指定的字符表达式，RIGHT( ) 函数从中返回最右边的若干字符。

nCharacters

指定要从字符串中返回的字符数目。如果 *nCharacters* 大于 *cExpression* 的长度，RIGHT( ) 函数返回整个字符表达式。如果 *nCharacters* 为负数或 0，

RIGHT( ) 函数返回空字符串。

说明

返回的字符从右边最后一个字符开始，包含指定数目的字符。

示例

```
CLEAR
? RIGHT('Redmond, WA', 2)  && 显示 WA
```

请参阅

A T ( ), LEFT( ), LTRIM( ) , RAT( ) , RTRIM( ) , SUBSTR( ) , TRIM( )

RIGHTC() 函数

从字符串中的最右边开始返回指定数目的字符。

语法

RIGHTC(cExpression, nCharacters)

返值类型

字符型

参数描述

cExpression

指定的字符表达式，RIGHT( ) 函数从中返回最右边的若干字符。

nCharacters

指定要从字符串中返回的字符数目。如果 *nCharacters* 大于 *cExpression* 的长度，RIGHT( ) 函数返回整个字符表达式。如果 *nCharacters* 为负数或 0，RIGHT( ) 函数返回空字符串。

说明

RIGHTC( ) 设计用于处理包含双字节字符的表达式。如果表达式中只有单字节字符，则 RIGHTC( ) 等同于 RIGHT( ) 。

返回的字符从右边最后一个字符开始，包含 nCharacters 数目的字符。

该函数适用于管理双字节字符集。

请参阅

AT_C( ), LEFTC( ) , RATC( ) , RIGHTC( ) , SUBSTRC( )

RightClick 事件

当用户在控件上按下并释放鼠标右键（鼠标辅键）时此事件发生。

语法

```
PROCEDURE Control.RightClick
    [LPARAMETERS nIndex]
```

参数描述

nIndex

唯一标识控件数组中的控件。

应用于

复选框，组合框，命令按钮，命令组，容器，控件，编辑框，表单，表格，标头，图像，标签，线条，列表框，选项按钮，选项组，页面，页框，形状，微调，文本框，工具栏

请参阅

Click 事件，DblClick 事件，DragDrop 事件，DragOver 事件，DropDown 事件，Enabled 属性，KeyPress 事件，MouseDown 事件，MouseMove 事件，MouseUp 事件，Scrolled 事件，Undock 事件

RightToLeft 属性

除非您运行的是中东版本的 Microsoft Windows，否则忽略。指定控件中文本的读顺序和表单的流输出显示。在设计时刻可见；在运行时刻读/写。

语法

Object.RightToLeft[= lExpression]

参数描述

lExpression

RightToLeft 属性的设置是:

设置	说明
真(.T.)	(默认)以从右到左的读顺序输入和显示控件中的文本。以从右到左的读顺序在表单上显示流输出(例如,用 ? 或 ?? 显示文本)。
假(.F.)	以从左到右的顺序输入和显示控件中的文本。以从左到右的读顺序在表单上显示流输出。

说明

用 Alignment 属性指定复选框和选项按钮中的文本方向。

应用于

复选框, 组合框, 命令按扭, 编辑框, 表单, 表格, 标签, 列表框, 选项按扭, 页框, _SCREEN, 微调, 文本框

请参阅

Alignment 属性

RLOCK() 函数

尝试给一个或多个表记录加锁。

语法

RLOCK([nWorkArea | cTableAlias]
| [cRecordNumberList, nWorkArea | cTableAlias])

返值类型

逻辑值

参数描述

nWorkArea | cTableAlias

指定表所在的工作区编号或表别名。如果不指定工作区或别名，则 RLOCK() 试图对当前选定工作区中表的当前记录加锁。

cRecordNumberList

指定 RLOCK() 尝试锁定的多个记录。字符表达式 *cRecordNumberList* 指定一个或多个要 RLOCK() 尝试锁定的记录的编号，多个记录的编号之间用逗号隔开。例如，要对表中前四个记录加锁，*cRecordNumberList* 应该为“1，2，3，4”。

要锁定多个记录，SET MULTILOCKS 必须设置为 ON，并且必须指定表所在的工作区号（*nWorkArea*）或表的别名（*cTableAlias*）。

要锁定多个记录，可以将记录指针移动到要锁定的记录上，执行 RLOCK() 或 LOCK() 命令将这个记录锁定，然后多次重复这个过程。

在 Visual FoxPro 中，可以指定 0 作为记录编号。指定 0 表示要锁定表头。

重要提示 锁定表头的时间要尽可能短。因为表头锁定期间，其他用户不能向表中添加记录。

对表头解锁可以用 UNLOCK RECORD 0 或 UNLOCK 或 UNLOCK ALL 等命令。

如果成功地锁定了所有 *cRecordNumberList* 之中的记录，则 RLOCK() 返回“真”(.T.)。

如果 *cRecordNumberList* 指定的记录有一个或多个不能加锁，则 RLOCK( ) 返回“假”(.F.)，并且对指定的任何记录都不加锁。以上两种情况中，原来已有的记录锁保持不变。
多个记录加锁是一个递增过程，即添加新的记录锁时并不释放原有的记录锁。
出于运行效率方面考虑，锁定整个表一般比锁定部分记录（哪怕是很少几个）要快。

说明

RLOCK( ) 等同于 LOCK( )。
如果加锁成功，则 RLOCK( ) 返回“真”(.T.)。此时，锁定的记录对于加锁的用户既可读也可写，而对于网络上的其他用户只读。
执行 RLOCK( ) 并不能保证每次加锁都成功。如果一个记录已由一个用户加锁，则其他用户不能对这个记录加锁；如果一个表已由一个用户加锁，则其他用户不能对这个表中的记录加锁。不管什么理由，只要记录不能加锁，RLOCK( ) 就返回“假”(.F.)。
默认情况下，RLOCK( ) 给记录加锁时只尝试一次。要想在第一次加锁尝试失败后自动重试加锁过程，应使用 SET REPROCESS 命令。SET REPROCESS 可以在加锁不成功的情况下控制尝试加锁的次数或尝试加锁的时间长度。有关 SET REPROCESS 和锁定表的详细内容，请参阅稍后的 SET REPROCESS 命令。
SET MULTILOCKS 命令决定能否锁定表中的多个记录。如果 SET MULTILOCKS 设置为 OFF （默认情况），则只能锁定表中的单个记录。如果 SET MULTILOCKS 设置为 ON，则可以锁定表中的多个记录。详细内容，请参阅稍后的 SET MULTILOCKS 命令。
已加锁的表记录只能由加锁的用户解锁。执行 UNLOCK 命令、关闭表或退出 Visual FoxPro 都可以对记录解锁。
UNLOCK 命令可以用来解开当前工作区、指定工作区或所有工作区内的记录锁。有

关详细内容，请参阅 UNLOCK。

将 SET MULTILOCKS 的设置由 ON 切换为 OFF 或由 OFF 切换为 ON 时，都隐含执行了 UNLOCK ALL 命令，所有工作区内的所有记录锁都被解除。

要关闭表，可以使用 USE、CLEAR ALL 或 CLOSE DATABASES 等命令。

有关网络中记录和文件锁定、共享表的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十七章“共享访问程序设计”。

示例

下面的示例对 customer 和 employee 表中的前四个记录进行锁定和解锁。

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
SET REPROCESS TO 3 AUTOMATIC
STORE '1,2,3,4' TO gcRecList
gcOldExc = SET('EXCLUSIVE')
SET EXCLUSIVE OFF
SELECT 0
USE employee   && 打开 Employee 表
SELECT 0
USE customer   &&打开 Customer 表
? LOCK('1,2,3,4', 'customer')   && 锁住 customer 表的前四个记录
? RLOCK(gcRecList, 'employee')   &&锁住 employee 表的前四个记录
UNLOCK IN customer
UNLOCK IN employee
SET EXCLUSIVE &gcOldExc
```

请参阅

CLEAR, CLOSE, FLOCK(), LOCK() , SET MULTILOCKS, SET RELATION,

SET REPROCESS ,UNLOCK,USE

_RMARGIN 系统变量

包含此变量是为了提供向后兼容性。可用报表设计器代替。

ROLLBACK 命令

取消当前事务期间所作的任何修改。

语法

ROLLBACK

说明

ROLLBACK 将表、表备注文件和索引文件恢复到事务开始前的状态。

当在事务中修改一个数据库中的记录时，网络上的其他用户在结束事务前不能访问（不论是读还是写）这些记录。

当网络上的其他用户要访问事务中修改过的记录时，必须等到事务结束之后。在可以

访问记录前，他们会收到“记录不可用...，请稍候”的信息。正因为如此,重要提示使事务持续的时间尽可能短或者在其他用户不需访问时再执行事务就特别重要。

ROLLBACK 恢复当前事务期间所作的任何修改。如果事务是嵌套的，则只恢复自前一个 BEGIN TRANSACTION 命令起所作的修改，恢复后，程序继续执行下面的语句。

如果事务处理期间给一个记录或文件加了锁，则 ROLLBACK 还要将这些锁将被解除。

示例

下面的示例打开了数据库 testdata 中的 customer 表。为 customer 表设置了最佳的表缓冲。显示了 cust_id 和 company 字段的内容，并且在缓冲数据中替换了 company 字段的内容。

用 BEGIN TRANSACTION 开始一个事务处理。TABLEUPDATE 用来写入更改到表中。显示新内容，再发出 ROLLBACK 命令以还原 company 字段的最初内容。重新显示 cust_id 和 company 字段，还有包含最初值的 company 字段。

```
CLOSE DATABASES
CLEAR

* Transactions are only supported within a DBC
OPEN DATABASE (HOME(2) + 'Data\testdata')
SET MULTILOCKS ON    && 要求缓冲

USE customer
=CURSORSETPROP("Buffering",5)
? 'The original company field'
LIST FIELDS cust_id, company NEXT 5
REPLACE ALL company WITH "***" && 改变字段内容

  BEGIN TRANSACTION
```

```
    =TABLEUPDATE(.T.)
    GO TOP
    ? 'The modified company field'
    LIST FIELDS cust_id, company NEXT 5
ROLLBACK            && 恢复原字段内容

=TABLEREVERT(.T.)
GO TOP
? 'The restored company field'
LIST FIELDS cust_id, company NEXT 5
```

请参阅

BEGIN TRANSACTION, END TRANSACTION, TXNLEVEL()

ROUND() 函数

返回圆整到指定小数位数的数值表达式。

语法

ROUND(nExpression, nDecimalPlaces)

返值类型

数值型

参数描述

nExpression

指定要圆整的数值表达式。

nDecimalPlaces

指定 *nExpression* 圆整到的小数位数。

如果 *nDecimalPlaces* 为负数，则 ROUND( ) 返回的结果在小数点左端包含 *nDecimalPlaces* 个零。例如，如果 *nDecimalPlaces* 为 -2，那么小数点左端的第一和第二个数字(个位和十位)均为 0。

说明

ROUND( )返回的值有 *nDecimalPlaces* 个小数位。ROUND( )忽略由 SET DECIMALS 指定的小数位。

示例

```
SET DECIMALS TO 4
SET FIXED ON       && 固定以小数方法显示
CLEAR

? ROUND(1234.1962, 3) && 显示 1234.1960
? ROUND(1234.1962, 2) && 显示 1234.2000
? ROUND(1234.1962, 0) && 显示 1234.0000
? ROUND(1234.1962, -1)  && 显示 1230.0000
? ROUND(1234.1962, -2)  && 显示 1200.0000
? ROUND(1234.1962, -3)  && 显示 1000.0000

SET FIXED OFF   && 恢复初始默认值
SET DECIMALS TO 2
```

请参阅

INT(), SET DECIMALS

ROW() 函数

包含向后兼容性。用 CurrentY 属性代替。

RowHeight 属性

指定表格控制中行的高度。设计时可用，运行时可读写。

语法

Grid.RowHeight[= nHeight]

参数描述

nHeight

指定表格控制中所有行的高度。

说明

如果 RowHeight 置为 –1（自动），则以 FontSize 属性的设置为标准自动调节行的大小。

应用于

表格

请参阅

FontSize 属性

RowSource 属性

指定组合框或列表框控制中值的来源。设计和运行时可用。

语法

Control.RowSource[= cName]

参数描述

cName

指定值的来源。

说明

值的来源可以是用逗号分隔的值列表、表、创建表或临时表的 SQL **语句、查询、数组、用逗号分隔的字段列表（可以在字段前加上一个句点和表的别名）、文件梗概（诸如** *.DBF **或** *.TXT）**、表的字段名称或菜单。使用** RowSourceType **属性可以设置行源的类型。**

在设计时可以使用 RowSource **属性给组合框或列表框控制指定多个列。要指定多个列，**

RowSourceType 应该设置为 1（值），然后按如下方式指定各个列（用逗号分隔）
Col1Row1,Col2Row1,Col1Row2,Col2Row2,,Col2Row3

用下面的语法可以为包含列的表指定一个别名：
Alias.Col1Row1,Col2Row1,Col1Row2,Col2Row2,,Col2Row3

指定的值按行填充控制，每行填充的列数由 ColumnCount 属性指定。上面的 example 假设 ColumnCount 的设置为 2。正如上面的 example 中表明的那样，在第 3 行第 1 列中将没有值，这是因为在第 3 行第 2 列之前有两个连续的逗号，逗号之间没有值。

示例

下面的 example 创建了一个列表框。出现在列表框中的项源是一个数组。数组的名称由 RowSource 属性指定。

RowSourceType 属性的设置是 5（数组）。它指定列表框中的项源是一个数组。

列表框的 MultiSelect 属性设置为“真”(.T.)，即允许从列表框中做多项选择。并用 ListCount 将从列表框中所做的选择显示出来。使用 Selected 和 List 属性可以确定列表框中项的数目和选择的项。

```
CLEAR

DIMENSION gaMyListArray(10)
FOR gnCount = 1 to 10   && 用字母填充数组
    STORE REPLICATE(CHR(gnCount+64),6) TO gaMyListArray(gnCount)
NEXT

frmMyForm = CREATEOBJECT('Form')   && 创建一个表单
frmMyForm.Closable = .f.   && 使控制菜单失效
```

```
frmMyForm.Move(150,10)   && 移动表单

frmMyForm.AddObject('cmbCommand1','cmdMyCmdBtn')  && 添加 “Quit”命令按钮
frmMyForm.AddObject('lstListBox1','lstMyListBox')  && 添加 ListBox 控制

frmMyForm.lstListBox1.RowSourceType = 5   && 指定一个数组
frmMyForm.lstListBox1.RowSource = 'gaMyListArray' && 包含列表框项的数组

frmMyForm.cmbCommand1.Visible =.T.   && 将 “Quit” 命令按钮设置为可见的
frmMyForm.lstListBox1.Visible =.T.   && 将 ListBox 设置为可见的

frmMyForm.SHOW   && 显示表单
READ EVEVTS   && 开始事件处理

DEFINE CLASS cmdMyCmdBtn AS CommandButton   && 创建命令按钮
    Caption = '\<Quit'   && 命令按钮上的标题
    Cancel = .T.   && 默认的取消按钮(Esc)
    Left = 125   && 命令按钮所在列
    Top = 210   && 命令按钮所在行
    Height = 25   && 命令按钮的高度

    PROCEDURE Click
       CLEAR EVENT   && 结束事件处理，关闭表单
       CLEAR   && 清除 Visual FoxPro 主窗口
ENDDEFINE

DEFINE CLASS lstMyListBox AS ListBox   && 创建 ListBox 控件
    Left = 10   && 列表框所在列
    Top = 10   && 列表框所在行
    MultiSelect = .T.   && 允许选择多个项

  PROCEDURE Click
```

```
    ACTIVATE SCREEN
    CLEAR
    ? "Selected items:"
    ? "--------------"
    FOR nCnt = 1 TO ThisForm.lstListBox1.ListCount
        IF ThisForm.lstListBox1.Selected(nCnt)   && 选择了此项吗?
            ? SPACE(5) + ThisForm.lstListBox1.List(nCnt) && 显示此项
        ENDIF
    ENDFOR

ENDDEFINE
```

应用于

组合框，列表框

请参阅

ColumnCount **属性**, RecordSource **属性**, RowSourceType **属性**

RowSourceType **属性**

指定控制中值的来源类型。设计和运行时可用。

语法

Control.RowSourceType[= nSource]

参数描述

nSource

RowSourceType 属性的设置有：

设置	说明
0	(默认值)无。如果使用了默认值，则在运行时使用 AddItem 或 AddListItem 方法填充列。
1	值。使用由逗号分隔的列填充。
2	别名。使用 ColumnCount 属性在表中选择字段。
3	SQL 语句。SQL SELECT 命令创建一个临时表或一个表。
4	查询(.QPR)。指定有 .QPR 扩展名的文件名。
5	数组。设置列属性可以显示多维数组的多个列
6	字段。用逗号分隔的字段列表。字段前可以加上由表别名和句点组成的前缀。
7	文件。用当前目录填充列。这时 RowSource 属性中指定的是文件梗概(诸如 *.DBF 或*.TXT) 或掩码。

续表

8	结构。由 RowSource 指定的表的字段填充列。 附注注意，当 RowSourceType 设置为 8 时，如果 RowSource 属性为空，则将当前选中的表用作组合框或列表框控件的源。否则，RowSource 属性指定了用作组合框或列表框控件的源表的别名。
9	弹出式菜单。包含此设置是为了提供向后兼容性。

有关使用每个 RowSourceType 设置的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十章“使用控件”中的“使用列表框和下拉列表框”。

示例

下面的 example 创建了一个列表框。列表框的项源是数组，数组的名称由 RowSource 属性指定。RowSourceType 属性的设置是 5（数组）。它指定列表框的项源是数组。

列表框的 MultiSelect 属性设置为“真”(.T.)，即允许从列表框中做多项选择。使用 ListCount、Selected 和 List 属性将列表框中所做的选择显示出来，从而可以确定列表框中项的数目和选择的项。

```
CLEAR

DIMENSION gaMyListArray(10)
FOR gnCount = 1 to 10   && 用字母填充数组
    STORE REPLICATE(CHR(gnCount+64),6) TO gaMyListArray(gnCount)
NEXT

frmMyForm = CREATEOBJECT('Form')   && 创建一个表单
frmMyForm.Closable = .f.   && 使控件菜单无效

frmMyForm.Move(150,10)   && 移动表单
```

```
frmMyForm.AddObject('cmbCommand1','cmdMyCmdBtn')  && 添加 “Quit” 命令按钮
frmMyForm.AddObject('lstListBox1','lstMyListBox')  && 添加 ListBox 控件

frmMyForm.lstListBox1.RowSourceType = 5  && 指定一个数组
frmMyForm.lstListBox1.RowSource = 'gaMyListArray' && 包含列表项的数组

frmMyForm.cmbCommand1.Visible =.T.  && 命令按钮可见
frmMyForm.lstListBox1.Visible =.T.  && 列表框可见

frmMyForm.SHOW  && 显示表单
READ 事件  && 开始事件处理

DEFINE CLASS cmdMyCmdBtn AS CommandButton  && 创建命令按钮
   Caption = '\<Quit'  && 命令按钮上的标题
   Cancel = .T.  && 默认的取消命令按钮(Esc)
   Left = 125  && 命令按钮所在的列
   Top = 210  && 命令按钮所在的行
   Height = 25  && 命令按钮的高度

   PROCEDURE Click
      CLEAR EVENT  && 结束事件处理，关闭表单
      CLEAR  && 清理 Visual FoxPro 主菜单
ENDDEFINE

DEFINE CLASS lstMyListBox AS ListBox  && 创建 ListBox 控件
   Left = 10  && List Box 所在的列
   Top = 10  && List Box 所在的行
   MultiSelect = .T.  && 允许多重选择

PROCEDURE Click
    ACTIVATE SCREEN
    CLEAR
```

```
    ? "Selected items:"
    ? "--------------"
    FOR nCnt = 1 TO ThisForm.lstListBox1.ListCount
        IF ThisForm.lstListBox1.Selected(nCnt)   && 选择了此项吗 ?
            ? SPACE(5) + ThisForm.lstListBox1.List(nCnt) && 显示项
        ENDIF
    ENDFOR

ENDDEFINE
```

应用于

组合框，列表框

请参阅

RowSource **属性**

RTOD() **函数**

将弧度转化为度。

语法

RTOD(nExpression)

返值类型

数值型

参数描述

nExpression
用弧度表示的数值表达式，RTOD( ) 将其转换成度。

说明

RTOD( ) 将表示弧度的数值表达式值转化成表示度的数值。

RTOD( ) 对于使用 Visual FoxPro 三角函数 COS( )、SIN( ) 和 TAN( ) 很有用。

要将度转化为弧度，应使用 DTOR。

示例

```
CLEAR
? RTOD(ACOS(0))  && 显示 90.00
STORE -1 to gnArcAngle
? RTOD(ACOS(gnArcAngle))  && 显示 180.00
? RTOD(ACOS(SQRT(2)/2)) && 显示 45.00
```

请参阅

COS( ), DTOR( ) , SIN( ) , TAN( )

RTRIM( ) 函数

删除了字符表达式续空格后，返回结果字符串。

语法

RTRIM(cExpression)

返值类型

字符型

参数描述

cExpression

指定要删除后续空格的字符表达式。

说明

RTRIM( ) 可用来确保删除用户输入到数据中的空格。RTRIM( )与 TRIM( ) 相同。

示例

```
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\testdata')
USE customer   && 打开 Customer 表
CLEAR
? 'The contact for '+ RTRIM(company) + ' is ' + contact
```

请参阅

ALLTRIM( ), LTRIM( ) , TRIM( )

RUN | ! 命令

运行外部操作命令或程序。

语法

RUN [/N] MS-DOSCommand | ProgramName

–或者–

! [/N] MS-DOSCommand | ProgramName

参数描述

MS-DOSCommand

指定要执行的 MS-DOS 命令。有关 MS-DOS 命令的详细内容，请参阅 MS-DOS 文档。

ProgramName

指定要运行的程序。可以指定基于 Windows 或基于 MS-DOS 的应用程序。

/N

指定 NOWAIT。包含 N 可以执行另一个基于 Windows 的应用程序。

说明

可以在命令窗口或在程序中执行 RUN 命令。

重要提示 要运行 RUN，操作系统文件 COMMAND.COM 必须在当前目录

中，或位于 MS-DOS COMSPEC 参数指定的地方。有关 COMSPEC 的详细内容，请参阅 MS-DOS 文档。

注意 请不要在 Visual FoxPro 内使用 RUN 来运行诸如 CHKDSK 这样的磁盘重组程序。这些程序修改磁盘上内容的方式可能会阻碍 Visual FoxPro 正常运行。

RUN and Visual FoxPro 如果 RUN 命令在 Visual FoxPro 以外运行程序，它要求程序与 FoxPro for MS-DOS 中的有一些细微的不同。.

如果在 RUN 中指定的程序没有扩展名，Visual FoxPro 首先在 MS-DOS 路径中寻找带有指定的名称的程序信息文件 (PIF)。PIF 允许在 Windows 下运行非 Windows 程序，并且可以指定程序的参数：程序是在窗口中运行还是全屏幕运行、分配给程序的内存数量等等。

如果找到了相应的 PIF，则使用 PIF 中指定的参数执行 PIF 中的程序；如果找不到 PIF，便在 MS-DOS 路径中按指定的名称搜索可执行程序。

当找不到 PIF 时，便使用安装在 Visual FoxPro 目录中的 PIF 文件 FOXRUN.PIF。FOXRUN.PIF 对程序在 Windows 中的运行做了一些配置。也可以修改 FOXRUN.PIF 来按另一种配置运行程序。

Foxrun.pif Foxrun.pif 允许在 Visual FoxPro 中执行基于 MS-DOS 和 Windows 的程序和命令。Foxrun.pif 与 Visual FoxPro 中的 vfp6.exe 必须在同一目录下。

/N 表示 NOWAIT。包含 N 可以执行其他基于 Windows 的应用程序。例如，下面的示例打开了 Windows 控制面板中的字符图：

```
! /N CHARMAP.EXE
```

下面的示例打开了 Windows 控制面板中的 Windows 颜色选取：

使用 RUN /N 或 ! /N 命令执行的 Windows 应用程序同通过程序管理器或文件管理器打开的应用程序运行的情况完全一样。可以使用 Windows 标准操作，在应用程序和 Visual FoxPro 或 FoxPro for Windows 之间来回切换。

可以在 /N 之后紧跟一个可选的数值，指定 Windows 应用程序的打开方式。注意不要在 /N 和数值之间加入空格。下表列出了有效的数值，同时说明了基于 Windows 的应用程序在每个数值下打开时对应的状态。

值	应用程序属性
1	活动且大小正常
2	活动且最小化
3	活动且最大化
4	不活动且大小正常
7	不活动且最小化

Running MS-DOS Programs in Visual FoxPro 默认情况下，Foxrun.pif 在窗口中运行指定的外部 MS-DOS 程序。当运行 MS-DOS 程序或命令时，窗口的标题是“FoxPro Run Command”。在 Visual FoxPro 中，当外部程序或命令结束执行时，即关闭“FoxPro Run Command”窗口。

Windows PIF 编辑器可以用来定制 Foxrun.pif。通过编辑 PIF 编辑器中的“退出时关闭窗口”复选框可以指定“Inactive FoxPro Run Command”窗口是继续保持打开还是关闭（Visual FoxPro 的默认情况）。还可通过选择 PIF 编辑器中的“全屏幕”复选框，将外部程序放在全屏状态下运行；也能为程序分配内存等。

Memory Considerations 默认情况下，Foxrun.pif 给外部命令或程序的运行分配最小的内存 256K。如果没有 256K 自由常规内存，Visual FoxPro 将会显示错误信息。要修正它，可试用下列一种或两种方法：

- 关闭应用程序和文件，释放更多的内存。
- 编辑 Foxrun.pif，减少“KB 需要值”文本框中所需的内存数量。

如果外部命令需要超过 256K 内存，MS-DOS 便在“FoxPro Run Command”窗口中

显示错误信息。要修正此错误，编辑 Foxrun.pif，增加“KB 需要值”文本框中所需的内存数。

请参阅

GETENV(),_SHELL

Run 事件

当一个 Active Document 运行用户代码时发生。

语法

```
PROCEDURE ActiveDoc.Run
LPARAMETERS cHyperlinkTarget
```

参数描述

cHyperlinkTarget

cHyperlinkTarget 是一个从 URL (Universal Resource Locator) 传递给 Run 事件的参数，当使用 HTTP (Hypertext Transfer Protocol) 运行一个 Active Document 时开始传递。该参数是字符值，并且使用井号 (#) 附加在 URL 中 Active Document 名称之后。例如，下列 URL 将“TargetString”字符串传递给名为 MyActiveDoc 的 Active Document 的 Run 事件：

```
Http://MyServer/MyActiveDoc.APP#TargetString
```

附注井号 (#) 是作为 cHyperlinkTarget 的第一个字符传递的。您可以在 Run 事件代码中使用 SUBSTR( ) 函数删除井号，如下所示：

```
LPARAMETERS cHyperLinkTarget
cNewTarget = SUBSTR(cHyperLinkTarget, 2, LEN(cHyperLinkTarget))
```

说明

Run 事件应该是您的 Active Document 应用程序的起点。通常，Run 事件包含的代码可以执行您的菜单代码，执行应用程序中的主表单，并且包含 READ EVENTS 以开始事件处理。

可以将启动代码放在 Active Document 的 Init 事件中，但是如果该代码过长，Active Document 的容器可能会产生一个超时错误。如果，不把启动代码放在 Init 事件中，则该代码就不应该需要用户的交互，也不需要创建用户界面。

应用于

ActiveDoc 对象

请参阅

Init 事件, SUBSTR( )

Run 方法

运行或预览项目中的一个文件。

语法

Object.Run( )

说明

如果成功运行或预览了该文件，则 Run 方法返回“真” (.T.)；否则返回“假”。使用 Run 方法可以预览标签或报表。

在运行或预览该文件之前发生 QueryRunFile 事件。如果在 QueryRunFile 事件中指定了 NODEFAULT，则不运行或预览该文件，并且 Run 方法返回“假” (.F.)。否则，运行或预览该文件，并且 Run 方法返回“真” (.T.)。

应用于

文件对象

请参阅

QueryAddFile 事件, QueryModifyFile 事件, QueryRunFile 事件, QueryRemoveFile 事件

_RUNACTIVEDOC 系统变量

指定一个启动 Active Document 的应用程序。

语法

_RUNACTIVEDOC = ProgramName

参数描述

ProgramName

指定用于启动 Active Document 的应用程序的名称。如果您的 Active Document 启动应用程序不在当前默认目录中，在应用程序名中应该包含路径。

说明

_RUNACTIVEDOC **包含一个应用程序的名称，该应用程序用于指定一个 Active Document 的启动选项。当在“工具”菜单中选择“运行 Active Document”命令时，就执行 _RUNACTIVEDOC 中指定的应用程序。在默认情况下，_RUNACTIVEDOC 包含 RUNACTD.PRG。**

也可以在“选项” 对话框的“文件位置”选项卡中，使用“ActiveDoc 启动程序”选项为 Active Document 指定启动应用程序。

可以按四种方式运行一个 Active Document：

- 使用 Visual FoxPro 运行时刻库，将 Active Document 包含在一个 Internet 浏览器中，例如 Microsoft Internet Explorer。
- 使用 Visual FoxPro 运行时刻库独立运行（无宿主）。
- 使用 Visual FoxPro 调试程序包含在一个 Internet 浏览器中。
- 使用 Visual FoxPro 调试程序独立运行。

请参阅

Active Documents, File Locations Tab, 选项对话框, SYS(4204)

_SAMPLES 系统变量

包含安装了 Microsoft Visual FoxPro 示例的路径。

语法

_SAMPLES[= *cPath*]

参数描述

*c*Path

指定包含了 Microsoft Visual FoxPro 示例的完整路径。

说明

在默认情况下，_SAMPLES 包含 Visual FoxPro 示例的目录。Visual FoxPro 示例是在进行“完全”安装 MSDN (Microsoft Developer’s Network) 时安装的。如果不安装 Visual FoxPro 示例，则 _SAMPLES 包含空字符串。

_SAMPLES 包含的路径是从 Windows 注册表中的 MSDN 注册关键字读取的。如果这个注册关键字不存在（例如，如果没有安装示例），则 _SAMPLES 包含空字符串。

也可以在“选项”对话框“文件位置”选项卡的“示例目录”项指定示例目录的路径。如使用“选项”对话框指定了示例目录的路径，则 _SAMPLES 包含所指定的路径。

HOME(2) 返回 _SAMPLES 中的路径。

请参阅

HOME( )