



返回总目录

STR () 函数

STRCONV () 函数

Stretch 属性

StrictDateEntry 属性

STRTOFILE () 函数

STRTRAN () 函数

STUFF () 函数

STUFFC () 函数

Style 属性

SUBSTR () 函数

SUBSTRC () 函数

SUM 命令

SUSPEND 命令

SYS () 函数概述

SYS (0) - 网络机器信息

SYS (1) - 儒略 (Julian) 系统日期

SYS (2) - 自午夜开始以秒计的时间

SYS (3) - 合法的文件名

SYS (5) - 默认驱动器

SYS (6) -当前打印设备
SYS (7) -当前格式文件
SYS (9) -Visual FoxPro 系列号
SYS (10) -来自儒略 (Julian)日期的字符串
SYS (11) -儒略 (Julian)日期
SYS (12) -可用内存的字节数
SYS (13) -打印机状态
SYS (14) -索引表达式
SYS (15) -字符转换
SYS (16) -执行程序文件名
SYS (17) -正在使用的处理器
SYS (18) -当前控件
SYS (20) -转换德文文本
SYS (21) -控制索引编号
SYS (22) -控制标志或索引名
SYS (23) -Visual FoxPro 的 EMS 内存用法
SYS (24) -EMS 内存限制
SYS (100) -控制台设置
SYS (101) -设备设置
SYS (102) -打印机设置
SYS (103) -对话设置

SYS (1001) – Visual FoxPro 内存
SYS (1016) – 用户对象的内存使用
SYS (1023) – 启用诊断帮助模式
SYS (1024) – 终止诊断帮助模式
SYS (1037) – ”页面设置”对话框
SYS (1269) – 属性信息
SYS (1270) – 对象位置
SYS (1271) – 对象的 SCX 文件
SYS (1272) – 对象层次
SYS (1500) – 激活系统菜单项
SYS (2000) – 文件名通配符匹配
SYS (2001) – SET 命令状态
SYS (2002) – 打开或关闭插入点
SYS (2003) – 当前目录
SYS (2004) – Visual FoxPro 启动目录或文件夹
SYS (2005) – 当前资源文件
SYS (2006) – 当前图形适配卡
SYS (2007) – 检查求和值
SYS (2010) – CONFIG.SYS 文件设置
SYS (2011) – 当前锁定状态
SYS (2012) – 备注字段块大小
SYS (2013) – 系统菜单名称字符串

- SYS (2014) – 最小化路径**
- SYS (2015) – 唯一过程名**
- SYS (2016) – SHOW GETS WINDOW 名称**
- SYS (2017) – 显示启动屏幕**
- SYS (2018) – 错误信息参数**
- SYS (2019) – 配置文件名称和位置**
- SYS (2020) – 默认磁盘空间**
- SYS (2021) – 筛选表达式**
- SYS (2022) – 磁盘簇 (块) 大小**
- SYS (2023) – 临时文件驱动器**
- SYS (2029) – 表类型**
- SYS (2333) – 开启或关闭 ActiveX 的双界面支持**

STR () 函数

返回与指定数值表达式对应的字符。

语法

STR (*nExpression* [, *nLength* [, *nDecimalPlaces*]])

返回值类型

字符型

参数描述

nExpression

STR () 要计算的数值表达式。

nLength

STR () 返回的字符串长度。该长度等于小数点和小数点右边每个数字所占字符的数目总和。

如果指定长度大于小数点左边数字位数，STR () 用前导空格填充返回的字符串；如果指定长度小于小数点左边的数字位数，STR () 返回一串星号，表示数值溢出。

如果 *nExpression* 是数值型或浮点型，并且 *nLength* 小于 *nExpression* 中数

字的个数，则 STR () 使用科学计数法。如果 nExpression 是一个整数，并且 nLength 小于 nExpression 中数字的个数，则 STR () 返回一个星号字符串，表明数字溢出。

如果不包含 nLength，则字符串的长度默认为 10 个字符。

nDecimalPlaces

由 STR () 返回的字符串中的小数位数，若要指定小数位数，必须同时包含 nLength。

如果指定的小数位数小于 nExpression 中的小数位数，则返回值类型四舍五入。

如果不包含 nDecimalPlaces，则小数点的位数默认为 0。

请参阅

VAL ()

STRCONV () 函数

将字符表达式转换成另一种形式。

语法

STRCONV (cExpression, nConversionSetting[, nLocaleID])

返回值类型

字符型

参数描述

cExpression

要 `STRCONV ()` 转换的字符表达式。

nConversionSetting

转换类型。下表列出 *nConversionSetting* 值和可执行的转换类型。

nConversion

转换

设置

1	将 <i>cExpression</i> 中的单字节转换为双字节
2	将 <i>cExpression</i> 中的双字节转换为单字节
3	将 <i>cExpression</i> 中的双字节 Hiragana 字符转换为双字节的 Katakana 字符
4	将 <i>cExpression</i> 中的双字节 Katakana 字符转换为双字节 Hiragana 字符。
5	将双字节转换为 UNICODE (宽字符)
6	将 UNICODE (宽字符) 转换为双字节
7	转换为与地区相关的小写形式
8	转换为与地区相关的大写形式

nLocaleID

指定用于转换的 *Locale ID*。如果 *nLocaleID* 无效，或计算机不支持，则造成“无效 Locale ID”错误。如果省略 *nLocaleID*，则在默认情况下使用系统的

Locale ID。

说明

该函数适用于管理双字节字符集。

请参阅

[CHRTRANC \(\)](#) , [IMESTATUS \(\)](#) , [ISLEADBYTE \(\)](#)

Stretch 属性

在一个控件内部，指定如何调整一幅图像以适应控件大小。设计和运行时可用。

语法

[Form.] Control.Stretch[= nType]

参数描述

nType

Stretch 属性的设置是：

设置

说明

0

(默认值) 剪裁。剪裁图像以适合控件。

续表

- 1 等比填充。调整图像大小以适合控件，同时保持图像的原始比例。
- 2 变比填充，调整图像大小以适合控件，但是不保持图像的原始比例。

应用于

图像，OLE 绑定型控件，OLE 容器控件

请参阅

[Autosize 属性](#)，[Sizable 属性](#)

StrictDateEntry 属性

指定在文本框中是否必须按特定的严格格式输入日期和日期时间值。
设计和运行时可用。

应用于

文本框

语法

`Object.StrictDateEntry[= nValue]`

参数描述

nValue

取下列设置之一：

设置

说明

0	<p>松散。可以按不严格的格式输入日期和日期时间值。由 <code>DateFormat</code> 属性或 <code>SET DATE</code> 命令决定输入日、月、年的顺序。空格、反斜杠、点号和当前日期分隔符（由 <code>DateFormat</code> 属性或 <code>SET DATE</code> 命令指定）可以用来分隔日期值。如果在日期中省略了年，则将当前年用于该日期。</p> <p>插入号（<code>^</code>）可以作为日期的第一个字符，以指定年-月-日的日期顺序，从而代替 <code>DateFormat</code> 属性或 <code>SET DATE</code> 命令指定的顺序。在日期时间值中，逗号或空格可以用来分隔日期和时间。当在日期时间值中只输入了小时时，如果在日期中包含年份，或者使用一个逗号分隔日期和时间，则可以省略冒号。</p> <p>注意 如果输入了一个无效的日期或日期时间值，则不显示一条错误信息，并且文本框的值设置为空的日期或日期时间值。您可以在 <code>Valid</code> 事件中检验日期是否有效。</p>
---	--

续表

- 1 (默认值) 严格。提供对 Visual FoxPro 以前版本的兼容性。日期和日期时间值中的日期必须按严格的 99/99/99 格式输入。如果 CENTURY 被设置为 ON，则必须按 99/99/9999 的格式。其中，99 分别代表日、月、年，9999 代表包含 CENTURY 的年。由 DateFormat 属性或 SET DATE 命令决定输入日、月、年的顺序。

说明

下表列出了当 StrictDateEntry 设置为 0 (松散) 时，可以在文本框中输入的有效日期和日期时间值。

日期或日期时间值

说明

12 31	当前年的 12 月 31 日。
12 31 98 14	1998 年 12 月 31 日下午 2 点。
12 31, 14	当前年的 12 月 31 日下午 2 点。
12 - 31 - 98, 2p	1998 年 12 月 31 日下午 2 点。注意，横线分隔符之间的空格。
^98-12-31, 2p	1998 年 12 月 31 日下午 2 点。插入符 (^) 指定了年-月-日的日期顺序，从而代替 DateFormat 属性或 SET DATE 命令指定的顺序。
^/12/31	当前年的 12 月 31 日。插入符 (^) 指定了年-月-日的日期顺序，从而代替 DateFormat 属性或 SET DATE 命令指定的顺序。

应用于

文本框

请参阅

Century 属性, DateFormat 属性, DateMark 属性, Hours 属性, Seconds 属性,
SET CENTURY, SET DATE, SET MARK TO, SET STRICTDATE

STRTOFILE () 函数

将一个字符串的内容写入一个文件。

语法

STRTOFILE (cExpression, cFileName [, lAdditive])

返回值类型

数值型

参数描述

cExpression

指定要写入一个文件的字符串。cExpression可以是一个原意字符串，几个结果为字符串的表达式，或者是一个字符型变量、数组元素或字段。

cFileName

指定要写入字符串的文件。如果该文件不在当前默认目录中，在文件名需要包含路径。如果所指定的文件不存在， Visual FoxPro 会自动创建它。

lAdditive

指定字符串是否追加于文件的末尾。如果 lAdditive 是”真” (.T.)，则该字符串追加于文件的末尾。

如果 lAdditive 是”假” (.F.) (默认值)，则使用该字符串覆盖文件的内容。

如果 SET SAFETY 设置为 ON，会询问您是否想要替换现有文件。如果 SET SAFETY 设置为 OFF，则在替换文件时不提示。

说明

STRTOFILE () 返回写入文件的字节数。

请参阅

FILETOSTR () , FWRITE ()

STRTRAN () 函数

在第一个字符表达式或备注字段中，搜索第二个字符表达式或备注字段，并每次用第三个字符表达式或备注字段替换第二个字符表达式或备注字段。

语法

STRTRAN (*cSearched*, *cSearchFor* [, *cReplacement*]
[, *nStartOccurrence*] [, *nNumberOfOccurrences*])

返回值类型

字符型

参数描述

cSearched

指定要在其中进行搜索的字符表达式，*cSearched* 参数可以是一个备注字段。

cSearchFor

在 *cSearched* 中要搜索的字符表达式，搜索过程区分大小写。*cSearchFor* 参数可以是一个备注字段。

cReplacement

用来替换在 *cSearched* 中每次出现的 *cSearchFor* 的字符表达式。如果省略 *cReplacement* 参数，则用空字符串替换每次出现的 *cSearchFor*。

nStartOccurrence

指定从哪次出现的 *cSearchFor* 开始替换。例如，如果 *nStartOccurrence* 是 4，则替换 *cSearched* 中第 4 次出现的 *cSearchFor*，而前面三次出现的 *cSearchFor* 保持不变；如果省略 *nStartOccurrence* 参数，替换开始的位置默认为第一次出现位置。

nNumberOfOccurrences

指定要替换的 *cSearchFor* 数目。如果省略了 *nNumberOfOccurrences*，则从用 *nStartOccurrence* 指定的位置开始，替换所有出现的 *cSearchFor*。

说明

指定替换开始的位置和替换次数，STRTRAN () 函数返回结果字符串。

示例

```
STORE 'abracadabra' TO gcString  
? STRTRAN ( gcString, 'a', 'z')  && 显示 "zbrzcdzbrz"  
? STRTRAN ( gcString, 'a', 'q', 2, 3)  && 显示 "abrqcqdqbra"
```

请参阅

[STUFF \(\)](#)

STUFF () 函数

返回一个字符串，此字符串是通过用另一个字符表达式替换现有字符表达式中指定数目的字符得到的。

语法

```
STUFF ( cExpression, nStartReplacement, nCharactersReplaced,  
cReplacement)
```

返回值类型

字符型

参数说明

cExpression

指定要在其中进行替换的字符表达式。

nStartReplacement

在 *cExpression* 中开始替换的位置。

nCharactersReplaced

要替换的字符数目。如果 *nCharactersReplaced* 是 0，则替换字符串

cReplacement 插入到 *cExpression* 中。

cReplacement

用以替换的字符表达式。如果 *cReplacement* 是空字符串，则从 *cExpression* 中删除用 *nCharactersReplaced* 指定的字符数目。

示例

```
STORE 'abcdefghijklm' TO gcString1
```

```
STORE '12345' TO gcString2
```

```
CLEAR
```

```
? STUFF ( gcString1, 4, 0, gcString2 )
```

&& 插入

```
? STUFF ( gcString1, 4, 3, gcString2 )
```

&& 替换

```
? STUFF ( gcString1, 4, 6, "" )
```

&& 删除

```
? STUFF ( gcString1, 4, 1, gcString2 )
```

&& 替换并插入

```
? STUFF ( gcString1, 4, 4, gcString2 )
```

&& 替换并删除

```
? STUFF ( gcString1, 4, LEN ( gcString1), gcString2 )
```

&& 替换，删除其余部分

请参阅

[LEFT \(\)](#) , [PADC \(\)](#) | [PADL \(\)](#) | [PADR \(\)](#) , [RIGHT \(\)](#) , [STRTRAN \(\)](#) ,
[SUBSTR \(\)](#)

STUFFC () 函数

返回一个字符串，此字符串是通过用另一个字符表达式替换现有字符表达式中指定数目的字符得到的。

语法

`STUFFC (cExpression, nStartReplacement, nCharactersReplaced,
cReplacement)`

返回值类型

字符型

参数描述

`cExpression`

指定要在其中进行替换的字符表达式。

`nStartReplacement`

指定 `cExpression` 中开始替换的位置。

`nCharactersReplaced`

指定要替换的字符数目。如果 `nCharacterReplaced` 是 0，则替换字符串 `cReplacement` 插入到 `cExpression` 中。

`cReplacement`

指定替换字符表达式。如果 `cReplacement` 是空字符串，则从 `cExpression` 中删除 `nCharactersReplaced` 指定的字符数目。

说明

STUFFC () 返回一个字符串，该字符串通过用另一个字符表达式替换现有字符表达式中指定数目的字符而得到。字符表达式可包含单字节和双字节字符的任意组合。该函数适用于管理双字节字符集。

请参阅

LEFTC () , RIGHTC () , STUFF () , SUBSTRC ()

Style 属性

指定控件的样式。设计和运行时可用。

语法

[*Form*.] Control.Style[= *nType*]

参数描述

nType

对于组合框控件，Style 属性的设置有：

设置

说明

- | | |
|---|--|
| 0 | 下拉组合框，包括一个下拉列表和一个编辑区。用户可以从列表中选择或在编辑区中输入字符。 |
| 2 | 下拉列表。用户必须从下拉列表中选择。 |

以下是选择哪种设置的准则：

- 将 Style 设置为 0（下拉组合框）可以给用户提供一个选择列表，同时也允许用户在编辑区中输入选择项。因为用户选择后将关闭列表部分，所以 Style 为 0 可以节省表单的空间。
- 将 Style 设置为 2（下拉列表）可以显示一个固定的选择列表，用户能够从中选择一项。当用户选择了一项后，将关闭列表部分。

对于复选框控件和选项按钮控件，Style 属性的设置有：

设置

说明

- | | |
|---|---|
| 0 | 标准方式。 |
| 1 | 图形方式。按钮看起来象一个命令按钮，可以包含图形和文本。当按钮两者都包含时，文本总是在按钮底部居中对齐的。 |

对于命令按钮控件，Style 属性的设置有：

设置

说明

- | | |
|---|-------|
| 0 | 标准方式 |
| 1 | 不可见方式 |

对于文本框控件，Style 属性的设置有：

设置

说明

-
- | | |
|---|--|
| 0 | (默认值) 正常方式 |
| 1 | 包含该设置是为了提供和 FoxPro 2.x 版本的 @...SAY 命令的向后兼容性。文本框是只读的并且不能接受焦点。 |

应用于

复选框，组合框，命令按钮，选项按钮，文本框

请参阅

[Click 事件](#)，[DbClick 事件](#)

SUBSTR () 函数

从给定的字符表达式或备注字段中返回指定数目的字符。

语法

SUBSTR (*cExpression*, *nStartPosition* [, *nCharactersReturned*])

返回值类型

字符型

参数描述

cExpression

指定要从其中返回字符串的字符表达式或备注字段。

StartPosition

指定返回的字符串在字符表达式或备注字段 *cExpression* 中的位置，*cExpression* 的第一个字符是位置 1。

注意：如果 TALK 设置为 ON，并且 *nStartPosition* 大于 *cExpression* 中的字符数目，那么 Visual FoxPro 产生错误信息；如果 TALK 设置为 OFF，那么返回一个空字符串。

NCharactersReturned

从 *cExpression* 中返回的字符数目。如果省略了 *nCharactersReturned* 参数，那么返回字符表达式的全部字符。

说明

SUBSTR () 从一个字符表达式或备注字段中返回一个字符串。该字符串起始于字符表达式或备注字段的指定位置，到指定数目字符结束。

当在一个 SELECT - SQL 命令中，将 SUBSTR () 用于备注字段时，可在 SUBSTR () 中使用 PADR () 函数，这样，将空备注字段或可变长度备注字段转化为字符串时，会保持一致结果。

当在调试窗口中发出 SUBSTR () 函数时，对于备注字段，SUBSTR () 不会返回值类型。要在调试窗口中返回值类型，可在 ALLTRIM () 函数中放入备注字段名，并且将 ALLTRIM () 函数放入 SUBSTR () 函数中。

示例

```
STORE 'abcdefghijklm' TO mystring  
CLEAR
```

? SUBSTR (mystring, 1, 5) && 显示 abcde
? SUBSTR (mystring, 6) && 显示 fghijklm

请参阅

AT () , LEFT () , PADR () , RIGHT () , STRTRAN () , STUFF ()

SUBSTRC () 函数

从给定的字符表达式或者备注字段返回字符串。

语法

SUBSTRC (*cExpression*, *nStartPosition* [, *nCharactersReturned*])

返回值类型

字符型

参数描述

cExpression

指定要从中返回字符串的字符表达式或备注字段。

NStartPosition

指定在字符表达式或备注字段 *cExpression* 中的位置，从该位置开始返回字符串。 *CExpression* 第一个字符的位置是 1。注意：如果 TALK 设置为 ON，

并且 *nStartPosition* 字符数比 *cExpression* 字符数大，Visual FoxPro 会产生错误信息；如果 TALK 设置为 OFF，则返回空字符串。

NCharactersReturned

指定从 *cExpression* 返回的字符数。如果省略 *nCharactersReturned*，那么返回字符表达式的全部字符。

说明

SUBSTRC () 从给定的字符表达式或备注字段中返回字符串。字符表达式或备注字段可以包含单字节和双字节字符的任何组合。在调试窗口使用 SUBSTRC () 时，SUBSTRC () 不能从备注字段返回值类型。若想在调试窗口返回一个值，需把备注字段放在 ALLTRIM () 函数中，再把 ALLTRIM () 放在 SUBSTRC () 函数中。该函数适用于管理双字节字符集。

请参阅

[AT_C \(\)](#) , [LEFTC \(\)](#) , [RIGHTC \(\)](#) , [STUFFC \(\)](#) , [SUBSTR \(\)](#)

SUM 命令

对当前选定表的指定数值字段或全部数值字段进行求和。

语法

SUM [*eExpressionList*]

[Scope] [FOR *lExpression1*] [WHILE *lExpression2*]
[TO *MemVarNameList* | TO ARRAY *ArrayName*]
[NOOPTIMIZE]

参数描述

eExpressionList

指定要总计的一个或多个字段或者字段表达式。如果省略字段表达式列表，则总计所有数值型字段。

Scope

指定要总计的记录范围。Scope 子句有：ALL、NEXT *nRecords*、RECORD 和 REST。其它信息，请参阅帮助中的“语句概览”的“Scope 子句”。

SUM 默认的作用域是所有记录（ALL）。

FOR *lExpression1*

指定对逻辑条件 *lExpression1* 值为“真”（.T.）的记录才进行总计。包含 FOR 子句可以有条件地总计记录，筛选出不想要的记录。

如果 *lExpression1* 是一个可优化表达式，那么 Rushmore 可以优化 SUM ... FOR 命令。为了获得最佳性能，请在 FOR 子句中使用可优化表达式。

其它相关信息，请参阅 [SET OPTIMIZE 命令](#) 和《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“使用 Rushmore 技术加速数据访问”。

WHILE *lExpression2*

指定一个条件，在当前表中只对逻辑表达式 *lExpression2* 为真（.T.）的记录

进行总计。

TO MemVarNameList

将每个总计值存入一个变量。如果 *MemVarNameList* 中指定的变量不存在，则 Visual FoxPro 自动创建。列表中的变量名用逗号分隔。

TO ARRAY ArrayName

将总计值存入变量数组中，如果在 SUM 命令中指定的数组不存在，则 Visual FoxPro 自动创建；如果数组存在但太小，不能包含所有的总计值，那么自动增加数组的大小以存放总计值。

NOOPTIMIZE

关闭 Rushmore 优化。

详细内容，请参阅前面的 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 ” 使用 Rushmore 技术加速数据访问 “ 。

示例

下列示例显示 porduct 表中， in stock 和 on_order 字段的总和，以及这两个值的和。

```
CLOSE DATABASES
OPEN DATABASE (HOME (2) + 'data\testdata')
USE products && 打开 Products 表
```

```
SUM in_stock, on_order, in_stock+on_order ;
  TO gnInStock, gnOnOrder, gnUnits
```

```
CLEAR
```

```
?
```

```
? 'Total in stock : ', gnInStock && 显示数值 3119.00
```

```
? 'Total on order: ', gnOnOrder && 显示数值 780.00  
? 'Total # units: ', gnUnits && 显示数值 3899.00
```

请参阅

[AVERAGE](#), [CALCULATE](#), [COUNT](#)

SUSPEND 命令

使用 SUSPEND 可暂停程序的执行，并返回到 Visual FoxPro 的交互状态。

语法

SUSPEND

说明

当程序暂停时，可以执行诸如检查变量值、打开跟踪窗口和调试窗口等命令。当程序暂停时，创建的所有变量都是 PRIVATE 变量。

请参阅

[CANCEL](#), [RESUME](#)

SYS () 函数概述

返回 Visual FoxPro 系统信息。

语法

SYS ()

返回值类型

字符型

说明

Visual FoxPro 的 SYS () 函数返回字符值，该字符值包含有用的系统信息。下表显示了每个 SYS () 函数返回的信息。

获得 SYS () 函数的帮助

- 在命令窗口或程序中突出 SYS () 函数，并按 F1 键。

函数	返回信息
SYS (0)	网络机器信息
SYS (1)	儒略 (Julian) 系统日期
SYS (2)	自午夜起计算的秒数
SYS (3)	合法的文件名

续表

SYS (5)	默认驱动器
SYS (6)	当前打印机设备
SYS (7)	当前格式文件
SYS (9)	Visual FoxPro 系列号
SYS (10)	来自儒略 (Julian) 日期的字符串
SYS (11)	儒略 (Julian) 日期
SYS (12)	可用内存的字节数
SYS (13)	打印机状态
SYS (14)	索引表达式
SYS (15)	字符转换

续表

SYS (16)	执行程序文件名
SYS (17)	正在使用的处理器
SYS (18)	当前控件
SYS (20)	转换德文文本
SYS (21)	控制索引编号
SYS (22)	控制标识或索引名
SYS (23)	Visual FoxPro 的 EMS 内存用法
SYS (24)	EMS 内存限制
SYS (100)	控制台设置
SYS (101)	设备设置

续表

SYS (102)	打印机设置
SYS (103)	对话设置
SYS (1001)	Visual FoxPro 内存
SYS (1016)	用户对象的内存使用
SYS (1023)	启用诊断帮助模式
SYS (1024)	终止诊断帮助模式
SYS (1037)	“页面设置”对话框
SYS (1269)	属性信息

SYS (1270)	对象位置
SYS (1271)	对象的 .SCX 文件
SYS (1272)	对象层次
SYS (1500)	激活 Visual FoxPro 系统菜单项
SYS (2000)	文件名通配符匹配
SYS (2001)	SET 命令状态
SYS (2002)	打开或关闭插入点
SYS (2003)	当前目录或文件夹
SYS (2004)	Visual FoxPro 启动目录或文件夹
SYS (2005)	当前资源文件
SYS (2006)	当前图形适配卡
SYS (2007)	检查求和值
SYS (2010)	CONFIG.SYS 文件设置
SYS (2011)	当前锁定状态

续表

SYS (2012)	备注字段块大小
SYS (2013)	系统菜单名称字符串
SYS (2014)	最小化路径
SYS (2015)	唯一的进程名
SYS (2016)	SHOW GETS WINDOW 名称
SYS (2017)	显示启动屏幕
SYS (2018)	错误信息参数
SYS (2019)	配置文件的名称和位置
SYS (2020)	默认磁盘空间
SYS (2021)	筛选表达式
SYS (2022)	磁盘簇 (块) 大小
SYS (2023)	临时文件驱动器
SYS (2029)	平台的指定路径转换
SYS (2333)	表类型
SYS (2334)	开启或关闭 ActiveX 的双界面支持

续表

SYS (2335)	自动服务启动
SYS (3004)	无人照管服务模式
SYS (3005)	返回环境 ID 值
SYS (3006)	设置环境 ID 值
SYS (3050)	设置语言和环境 ID 值
SYS (3051)	设置缓冲内存大小
SYS (3052)	忽略 SET REPROCESS 锁定
SYS (3053)	ODBC 环境句柄
SYS (3054)	显示 Rushmore 优化的级别
SYS (3055)	FOR 和 WHERE 子句的复杂性
SYS (3056)	读取注册设置
SYS (4204)	Active Document 调试支持

SYS (0) -网络机器信息

当在网络环境中使用 Visual FoxPro 时，SYS (0) 返回网络机器信息。

语法

SYS (0)

返回值类型

字符型

说明

必须用网络软件指定机器信息，并且必须加载网络外壳程序。

如果没有指定机器信息，或者没有加载网络外壳程序，SYS (0) 返回一个字符串。该字符串包括 15 个空格，一个数字符号 (#)，后接另一个空格，最后是 0。关于定义机器信息的详细内容，请参考您的网络文档。

在独立环境下使用 Visual FoxPro 时，SYS (0) 返回 1。

SYS (0) 返回机器名称和用户名称。

请参阅

[SYS \(\) 函数概述](#)

SYS (1) - 儒略 (Julian) 系统日期

以儒略 (Julian) 日期字符串的形式返回当前系统日期。

语法

SYS (1)

返回值类型

字符型

说明

SYS (1) 的返回值类型在美国有效。在 Visual FoxPro 版本中，可表示的日期范围是 1752 年 9 月 14 日 — 9999 年 12 月 31 日。

示例

```
? SYS ( 1 )  
? SYS ( 10, VAL ( SYS ( 1 )))
```

请参阅

[CDOW \(\)](#) , [CTOD \(\)](#) , [DATE \(\)](#) , [DATETIME \(\)](#) , [DAY \(\)](#) , [DOW \(\)](#) ,
[DTC \(\)](#) ,
[SYS \(\) 函数概述](#) , [SYS \(10\)](#) , [SYS \(11\)](#)

SYS (2) -自午夜开始以秒计的时间

返回自午夜零点开始以来的时间，按秒计算。

语法

SYS (2)

返回值类型

字符型

示例

? TIME ()
? SYS (2)

请参阅

[DATETIME \(\)](#) , [SECONDS \(\)](#) , [SYS \(\)](#) 函数概述 , [TIME \(\)](#)

SYS (3) -合法的文件名

返回一个合法文件名，可用来创建临时文件。

语法

SYS (3)

返回值类型

字符型

说明

在一台速度很快的计算机上，当成功地发出 SYS (3) 时，可能返回一个不唯一的名字。使用 SUBSTR (SYS (2015), 3, 10) 可以创建唯一的、合法的、8 个字符的文件名。

请参阅

[FILE \(\)](#) , [SYS \(\) 函数概述](#) , [SYS \(2015 \)](#)

SYS (5) - 默认驱动器

返回当前 Visual FoxPro 的默认驱动器。

语法

SYS (5)

返回值类型

字符型

说明

这个函数返回当前 Visual FoxPro 的默认驱动器，使用 SET DEFAULT 来指定一个默认驱动器。

示例

```
IF _DOS OR _WINDOWS
    SET DEFAULT TO C:
ENDIF
? SYS ( 5 )
```

请参阅

[DRIVETYPE \(\)](#) , [SET DEFAULT](#) , [SYS \(\)](#) 函数概述

SYS (6) - 当前打印设备

返回当前打印设备。

语法

SYS (6)

返回值类型

字符型

说明

这个函数返回当前的 SET PRINTER TO 设置值。

示例

```
? SYS ( 6 )  
SET PRINTER TO output.txt  
? SYS ( 6 )
```

请参阅

[SET PRINTER, SYS \(\) 函数概述](#)

SYS (7) -当前格式文件

返回当前格式文件的名称。

语法

SYS (7 [, *nWorkArea*])

返回值类型

字符型

参数描述

nWorkArea

指定 SYS (7) 要返回的格式文件名称的工作区编号。如果省略 *nWorkArea* 参数，则返回当前工作区的格式文件名称。

说明

用 SET FORMAT 打开格式文件。在指定的工作区中，如果没有打开的格式文件，SYS (7) 返回一个空串。

请参阅

SET FORMAT, SYS () 函数概述

SYS (9) -Visual FoxPro 系列号

返回 Visual FoxPro 的系列号。

语法

SYS (9)

返回值类型

字符型

请参阅

[SYS \(\) 函数概述](#), [VERSION \(\)](#)

SYS (10) -来自儒略 (Julian)日期的字符串

将儒略 (Julian) 日期转换成一个字符串。

语法

SYS (10, *nJulianDayNumber*)

返回值类型

字符型

说明

`SYS (10)` 从 Julian 日期 *nJulianDayNumber* 中返回一个字符型日期。

示例

```
? SYS ( 1 )  
? SYS ( 10, VAL ( SYS ( 1 )))
```

请参阅

`CDOW ()` , `CTOD ()` , `DATE ()` , `DATETIME ()` , `DAY ()` , `DOW ()` ,
`DTOC ()` ,

`SYS ()` 函数概述, `SYS (1)`, `SYS (11)`

`SYS (11)` - 儒略 (Julian) 日期

将日期格式的表达式或字符串转换成儒略 (Julian) 日期。

语法

`SYS (11, dExpression | tExpression | cExpression)`

返回值类型

字符型

说明

从日期表达式 *dExpression*、日期时间表达式 *tExpression* 或日期格式的字符串 *cExpression* 中，SYS (11) 返回儒略 (Julian) 日期，该日期号作为字符串返回。

示例

```
? SYS ( 11, {^1998-06-06})  
? SYS ( 11, '06/06/1998')
```

请参阅

[CDOW \(\)](#) , [CTOD \(\)](#) , [DATE \(\)](#) , [DATETIME \(\)](#) , [DAY \(\)](#) , [DOW \(\)](#) ,
[DTC \(\)](#) ,
[SYS \(\) 函数概述](#) , [SYS \(10\)](#)

SYS (12) - 可用内存的字节数

返回 640K 以下、可用于执行外部程序的内存数量。

语法

SYS (12)

返回值类型

字符型

说明

在 Visual FoxPro 中，SYS (12) 总是返回 655,360。

SYS (12) 与 MEMORY () 相似，但有两点不同：

- SYS (12) 以字节为单位返回可用内存的数量，MEMORY () 以千字节 (KB) 为单位返回可用内存数量。
- SYS (12) 返回一个字符串，而 MEMORY () 返回一个数值。

请参阅

[MEMORY \(\)](#), [RUN |!](#), [SYS \(\) 函数概述](#), [SYS \(23\)](#), [SYS \(24\)](#), [SYS \(1001\)](#), [SYS \(1016\)](#)

SYS (13) - 打印机状态

返回打印机的状态。

语法

SYS (13)

返回值类型

字符型

说明

如果打印机没有准备就绪，SYS (13) 返回 OFFLINE；如果打印机准备就绪，SYS (13) 返回 READY。

如果打印机连接到一个串行通信端口，当打印机返回 “Clear To Send Data” 或 “Data

Set Ready” 时，SYS (13) 返回 READY。

如果打印机连接到一个并行端口上，当打印机返回 O
Error” 、”TimeOut” 、”Printer Busy” 或”Printer Not Selected” 时，SYS (13) 返回
OFFLINE。

示例

```
IF SYS (13) = 'OFFLINE'  
    WAIT WINDOW 'Printer is offline'  
ENDIF
```

请参阅

[PRINTSTATUS \(\)](#) , [SET PRINTER](#) , [SYS \(\)](#) [函数概述](#)

SYS (14) -索引表达式

返回一个打开的、 单项索引文件的索引表达式， 或者返回复合索引文件中索引标识的索引表达式。

语法

SYS (14, *nIndexNumber* [, *nWorkArea* | *cTableAlias*])

返回值类型

字符型

参数描述

nIndexNumber

从打开的索引文件或标识中，指定返回哪一个索引表达式。当 *nIndexNumber* 从 1 递增到打开的单项索引文件标识、结构复合索引标识、独立复合索引标识的总和时，SYS (14) 以下面的顺序从打开的索引文件和标识中返回索引表达式：

1. 首先从单项索引文件（如果都是打开的话）中返回索引表达式。包含在 USE 或 SET INDEX 中的单项索引文件的顺序决定索引表达式返回的顺序。
2. 接着返回结构复合索引（如果存在的话）中每个标识的索引表达式。以标识在结构索引中的创建顺序返回索引表达式。
3. 最后返回打开的独立复合索引中每个标识的索引表达式。以标识在独立复合索引中的创建顺序返回索引表达式。

如果 *nIndexNumber* 大于打开的单项索引文件、结构复合索引标识、独立复合索引标识的总和时，SYS (14) 返回一个空串。

nWorkArea | cTableAlias

指定工作区编号或工作区别名。如果省略了 *nWorkArea* 和 *cTableAlias* 参数，那么从当前工作区内打开的索引文件中返回索引表达式；如果表没有指定的别名，Visual FoxPro 将产生错误信息。

说明

当用 INDEX 创建索引文件或标识时，已经指定了一个索引表达式。当用索引文件或

标识对表进行排序时，索引表达式决定如何显示和访问表。

有关索引表达式和创建索引文件和标识的详细内容，请参阅 INDEX。SYS (14) 与 KEY () 函数相似。

USE 和 SET INDEX 命令都支持索引文件名列表，该列表可以为一个表打开索引文件。单项索引文件、结构复合索引文件或独立的复合索引文件名的任意组合都可以包含到索引文件列表中。

请参阅

[CDX \(\)](#) , [KEY \(\)](#) , [INDEX](#) , [MDX \(\)](#) , [NDX \(\)](#) , [REINDEX](#) , [SET INDEX](#) , [SET ORDER](#) , [SYS \(\) 函数概述](#) , [TAG \(\)](#)

SYS (15) - 字符转换

从第一个字符串中变换出第二个字符串，包含此函数是为了提供向后兼容性。可使用 SET COLLATE 来代替。

SYS (16) - 执行程序文件名

返回正在执行的程序文件名。

语法

`SYS (16 [, nProgramLevel])`

返回值类型

字符型

参数描述

`nProgramLevel`

指示从第几层嵌套中获取程序名。该值范围可以从 1 变化到程序嵌套的层数。

如果 `nProgramLevel` 是 0 或 1，`SYS (16)` 返回主程序名（第一个执行的程序）；如果省略了 `nProgramlevel` 参数，那么返回当前正在执行的程序名；如果 `nProgramlevel` 参数大于程序嵌套深度，则返回一个空串。

说明

如果要从错误中恢复过来，该函数很有用。`SYS (16)` 与 `PROGRAM ()` 相似。然而，`SYS (16)` 返回带路径的程序名，而 `PROGRAM ()` 只返回程序名。

如果正在执行一个过程或函数，`SYS (16)` 在过程或函数名之后返回包含该过程或函数的文件名。当正在执行的程序是应用程序（.APP）的一部分，那么 `SYS (16)` 只返回程序名（路径不和程序名一起返回）。

示例

在下面的小程序示例中返回程序嵌套数目：

```
STORE 1 TO gnX  
DO WHILE LEN ( SYS ( 16,gnX) ) != 0
```



```
? SYS ( 16,gnX)
STORE gnX+1 TO gnX
ENDDO
```

请参阅

[LINENO \(\)](#) , [ON ERROR, PROGRAM \(\)](#) , [SYS \(\)](#) 函数概述

SYS (17) -正在使用的处理器

返回正在使用的中央处理器 (CPU) 。

语法

SYS (17)

返回值类型

字符型

说明

该函数返回正在使用的中央处理单元类型：80386，80486 …… 等等。

示例

```
? 'Processor in use: ', SYS ( 17)
```

请参阅

[SYS \(\)](#) 函数概述

SYS (18) - 当前控件

以大写字母形式返回用于创建当前**控件**的变量、数组元素或字段的名称。包含该函数是为了提供向后兼容性，可使用 ActiveControl 属性代替

SYS (20) - 转换德文文本

将包含德文文本的字符表达式转换成一个字符串。包含该函数是为了提供向后兼容性。可使用 SET COLLATE 代替。

SYS (21) - 控制索引编号

对于当前所选工作区中的主控制 .CDX 复合索引标识或 .IDX 单项索引文件，以字符串形式返回其索引位置编号。

语法

SYS (21)

返回值类型

字符型

说明

索引位置编号由 USE 和 SET INDEX 命令指定的 .IDX 单项索引文件和 .CDX 复合索引标识的顺序来决定。

可以使用 SET INDEX、SET ORDER 和 USE 命令指定哪一个 .IDX 单项索引文件或 .CDX 复合索引标识是主控制索引文件或标识。有关指定主控制索引或标识的详细内容，请参阅 SET INDEX 和 USE。

如果没有主控制 .CDX 复合索引标识或 .IDX 索引文件（例如，发出 SET ORDER TO 命令以记录的自然顺序显示和访问表），那么 SYS (21) 返回"0"。

请参阅

CDX () , KEY () , INDEX , MDX () , NDX () , REINDEX , SET INDEX , SET ORDER , SYS () 函数概述 , SYS (22) , TAG () , USE

SYS (22) - 控制标志或索引名

返回表的主控制 .CDX 复合索引标识或 .IDX 单项索引文件的名称。

语法

SYS (22 [, *nWorkArea*])

返回值类型

字符型

参数描述

nWorkArea

表的工作区编号。SYS (22) 返回该表的主控 .CDX 复合索引标识或 .IDX 单项索引文件的名称。

说明

可以使用 SET INDEX、SET ORDER 和 USE 命令来指定哪一个 .IDX 单项索引文件或 .CDX 复合索引标识是主控制索引文件或标识。有关指定主控制索引或标识的详细内容，请参阅 SET INDEX、SET ORDER 和 USE。

如果没有主控 .CDX 复合索引标识或 .IDX 索引文件（例如，发出 SET ORDER TO 命令，以记录的自然顺序显示和访问表），那么 SYS (22) 返回一个空串。

请参阅

[CDX \(\)](#), [KEY \(\)](#), [INDEX](#), [MDX \(\)](#), [NDX \(\)](#), [REINDEX](#), [SET INDEX](#), [SET ORDER](#), [SYS \(\) 函数概述](#), [SYS \(21\)](#), [TAG \(\)](#)

SYS (23) - Visual FoxPro 的 EMS 内存用法

返回 FoxPro for MS-DOS 标准版当前正在使用的 EMS 内存数量 (以 16K 段形式) 。

语法

SYS (23)

返回值类型

字符型

说明

如果没有使用 EMS 内存，SYS (23) 返回 0。

在 FoxPro for MS-DOS、Visual FoxPro、FoxPro for Windows 和 FoxPro for Macintosh 扩展版 (32 位) 中，SYS (24) 总是返回 0。

请参阅

[SYS \(\) 函数概述](#), [SYS \(12 \)](#), [SYS \(24 \)](#), [SYS \(1001 \)](#), [SYS \(1016 \)](#)

SYS (24) - EMS 内存限制

返回 FoxPro for MS-DOS 配置文件中关于 EMS 限制的设置值。

语法

SYS (24)

返回值类型

字符型

说明

在 FoxPro for MS-DOS 标准版的配置文件中，如果不包含 EMS 限制，那么 SYS (24) 返回 0。

在 FoxPro for MS-DOS 和 Visual FoxPro 的 32 位扩展版中，SYS (23) 总是返回 0。

其他相关信息，请参阅帮助中的“安装指南”的第 3 章“Visual FoxPro 配置”。

请参阅

[SYS \(\) 函数概述](#), [SYS \(12\)](#), [SYS \(23\)](#), [SYS \(1001\)](#), [SYS \(1016\)](#)

SYS (100) -控制台设置

返回当前 SET CONSOLE 的设置。

语法

SYS (100)

返回值类型

字符型

示例

```
IF SYS ( 100 ) != 'ON'  
    SET CONSOLE ON  
ENDIF
```

请参阅

[SET CONSOLE, SYS \(\) 函数概述](#)

SYS (101) - 设备设置

返回当前 SET DEVICE 的设置。

语法

SYS (101)

返回值类型

字符型

示例

```
IF SYS ( 101 ) != 'SCREEN'  
    SET DEVICE TO SCREEN  
ENDIF
```

请参阅

[SET DEVICE, SYS \(\) 函数概述](#)

SYS (102) -打印机设置

返回当前 SET PRINTER 的设置。

语法

SYS (102)

返回值类型

字符型

示例

```
IF SYS ( 102 ) != 'OFF'  
    SET PRINTER OFF  
ENDIF
```

请参阅

[SET PRINTER, SYS \(\) 函数概述](#)

SYS (103) -对话设置

返回当前 SET TALK 的设置。

语法

SYS (103)

返回值类型

字符型

示例

```
IF SYS ( 103 ) != 'ON'  
    SET TALK ON  
ENDIF
```

请参阅

[SET TALK, SYS \(\) 函数概述](#)

SYS (1001) - Visual FoxPro 内存

返回 Visual FoxPro 内存管理器可用的内存总数。

语法

SYS (1001)

返回值类型

字符型

说明

在 Visual FoxPro 中，SYS (1001) 返回虚拟内存池的大小，该大小约是物理内存数量的四倍。

请参阅

[SYS \(\) 函数概述](#), [SYS \(12 \)](#), [SYS \(23 \)](#), [SYS \(24 \)](#), [SYS \(1016 \)](#)

SYS (1016) - 用户对象的内存使用

返回用户自定义对象所使用的内存数量。

语法

SYS (1016)

返回值类型

字符型

说明

用户自定义对象包括：用户自定义的窗口、菜单栏、菜单、变量、数组、打开的表、用低级文件函数打开的文件等等。

请参阅

[SYS \(\) 函数概述](#), [SYS \(12 \)](#), [SYS \(23 \)](#), [SYS \(24 \)](#), [SYS \(1001 \)](#)

SYS (1023) - 启用诊断帮助模式

启用诊断帮助模式，能够俘获传递给 Visual FoxPro 帮助系统的 HelpContextID 值。

语法

SYS (1023)

返回值类型

字符型

说明

对于调试应用程序的自定义帮助系统，SYS (1023) 很有用，它返回空字符串。

当用 SYS (1023) 启用诊断模式之后，无论在什么时候按 F1 键或发出 HELP 命令，都会显示一个对话框。该对话框显示出要传递给 Visual FoxPro 帮助系统的 HelpContextID 值，并且给出选项，确定是否将 HelpContextID 传递给 Visual FoxPro 帮助系统。如果选择“是”，则将 HelpContextID 传递给 Visual FoxPro 帮助系统，并且显示相应的帮助主题（如果 HelpContextID 可用的话）；如果选择了“否”，则不将 HelpContextID 传递给 Visual FoxPro 帮助系统，也就不会显示相应的帮助主题。

用 SYS (1024) 终止诊断帮助模式，并且恢复默认的 Visual FoxPro 帮助系统处理方式。

请参阅

[HELP, HelpContextID 属性, SET HELP, SYS \(1024\)](#)

SYS (1024) - 终止诊断帮助模式

终止 SYS (1023) 所启用的诊断帮助模式。

语法

SYS (1024)

返回值类型

字符型

说明

使用 SYS (1024) 可以终止 SYS (1023) 启用的诊断帮助模式，并且还原成默认的 Visual FoxPro 帮助系统处理方式。SYS (1023) 可用来调试应用程序的自定义帮助系统。SYS (1024) 返回空字符串。

请参阅

[HELP, HelpContextID 属性, SET HELP, SYS \(1023\)](#)

SYS (1037) - “ 页面设置 ” 对话框

显示“页面设置”对话框。

语法

SYS (1037)

返回值类型

字符型

说明

SYS (1037) 显示“页面设置”对话框，允许用户调整打印机设置（诸如打印纸张大小和走纸方向）。在该对话框中可以调整的设置依赖于所安装的打印机。SYS (1037) 总是返回空串。

请参阅

[GETPRINTER \(\)](#) , [PRINTSTATUS \(\)](#) , [SET DEVICE](#) , [SET PRINTER](#)

SYS (1269) - 属性信息

返回一个逻辑值，该值代表属性的默认值是否更改或属性是否为只读。

语法

SYS (1269, *oObjectName*, *cProperty*, *nPropertyAttribute*)

返回值类型

逻辑值

参数描述

oObjectName

指定一个对象，其属性信息被返回。

cProperty

指定一个属性，SYS (1269) 返回其信息。

nPropertyAttribute

指定属性状态。当 *nPropertyAttribute* 是 0 时，若属性默认值被更改，则 SYS (1269) 返回“真” (.T.); 否则，返回“假” (.F.)。当 *nPropertyAttribute* 是 1 时，若属性是只读的，则 SYS (1269) 返回“真” (.T.); 若属性是读写的，返回“假” (.F.)。

说明

程序中 在设置属性之前，SYS (1269) 确定属性默认值被更改或属性为只读。

请参阅

`CREATE FORM`, `GETPEM ()`, `PEMSTATUS ()`, `SYS (1270)`, `SYS (1271)`, `SYS (1272)`

SYS (1270) - 对象位置

返回对指定位置对象的引用。

语法

`SYS (1270 [, nXCoord, nYCoord])`

返回值类型

对象型，逻辑型

参数描述

nXCoord

指定水平坐标。

nYCoord

指定垂直坐标。

说明

如果省略坐标 *nXCoord* 和 *nYCoord*，将返回对当前鼠标指针下对象的引用；如果包含坐标 *nXCoord* 和 *nYCoord*，将返回对指定位置对象的引用。

如果在鼠标指针或 nXCoord 和 nYCoord 指定的坐标下没有对象，则返回“假”（.F.）。

请参阅

AMOUSEOBJ（），CREATE FORM，GETPEM（），PEMSTATUS（），
SYS（1269），SYS（1271），SYS（1272）

SYS（1271）- 对象的 SCX 文件

返回 .SCX 文件名，该文件存储指定的实例对象。

语法

SYS（1271，*ObjectName*）

返回值类型

字符型，逻辑型

参数描述

oobjectName

指定对象名，此函数将返回存储它的 .SCX 文件名。

说明

SYS（1271）返回 .SCX 文件名，该文件存储指定实例对象。

如果在 .SCX 文件中找不到指定的对象，则 SYS（1271）返回逻辑值“假”（.F.）（例

如，在命令窗口中创建对象)。

请参阅

[CREATE FORM](#), [GETPEM \(\)](#), [PEMSTATUS \(\)](#), [SYS \(1269\)](#), [SYS \(1270\)](#), [SYS \(1272\)](#)

SYS (1272) - 对象层次

返回指定对象的对象层次。

语法

`SYS (1272, oObjectName)`

返回值类型

字符型

参数描述

`oObjectName`

指定对象名，并返回该对象的层次。

请参阅

[CREATE FORM](#), [GETPEM \(\)](#), [PEMSTATUS \(\)](#), [SYS \(1269\)](#), [SYS](#)

(1270), SYS (1271)

SYS (1500) -激活系统菜单项

激活一个 Visual FoxPro 系统菜单项。

语法

SYS (1500, *cSystemItemName*, *cMenuName*)

返回值类型

字符型

参数描述

cSystemItemName

指定希望被激活得 Visual FoxPro 系统菜单项。

cMenuName

指定系统菜单项 *cSystemItemName* 所在的菜单或子菜单。

说明

如果您想查看 Visual FoxPro 的所有菜单和菜单项，可以使用 SYS (2013) 显示系统菜单的列表。

您不能使用 SYS (1500) 激活用户自定义菜单项，或被禁止 (disabled) 的系统菜单。

SYS (1500) 总是返回空字符串。

示例

下面的示例用 SYS (1500) 粘贴一个命令到程序文件中。

```
_CLIPTEXT = "MESSAGEBOX ('TEST')" && 要粘贴的命令  
MODIFY COMMAND myprog NOWAIT && 打开程序文件  
SYS (1500, '_MED_PASTE', '_MEDIT') && 粘贴菜单项
```

请参阅

[SYS \(2013\), System Menu Names](#)

SYS (2000) - 文件名通配符匹配

返回与文件名梗概匹配的第一个文件名。

语法

```
SYS (2000, Skeleton [, 1])
```

返回值类型

字符型

参数描述

Skeleton

指定文件名梗概。文件名梗概可以包含通配符”?”和”*”。

1

返回下一个匹配文件的文件名。

说明

如果找不到匹配文件，SYS (2000) 返回空串。

示例

```
? SYS ( 2000, 'FOX.*')  
? SYS ( 2000, 'FOX.*', 1)
```

请参阅

[LOCFILE \(\)](#) , [SYS \(\)](#) [函数概述](#)

SYS (2001) - SET 命令状态

返回指定的 SET 命令的状态。

语法

```
SYS ( 2001, cSET Command [, 1 | 2])
```

返回值类型

字符型

参数描述

cSETCommand

指定 SET 命令，SYS (2001) 返回该 SET 命令的状态。

1 | 2

一些 SET 命令有两个或两个以上的设置。例如，SET PRINTER ON，SET PRINTER OFF 和 SET PRINTER TO *FileName*。不带 1 或 2 参数使 SYS (2001) 返回 ON 或 OFF 设置，带 1 或 2 参数使 SYS (2001) 返回这些命令的附加的设置。请参阅 SET () 函数中的 SET 命令列表。

说明

SYS (2001) 等同于 SET () 函数。

示例

```
? SYS (2001,'PRINTER')  
? SYS (2001,'PRINTER',1)
```

请参阅

[SET \(\) , SYS \(\) 函数概述](#)

SYS (2002) -打开或关闭插入点

打开或关闭插入点。

语法

SYS (2002 [, 1])

返回值类型

无

说明

SYS (2002) 关闭插入点，SYS (2002, 1) 打开插入点。

有关打开或关闭插入点的详细内容，请参阅 SET CURSOR。

请参阅

[INKEY \(\)](#) , [SET CURSOR](#) , [SYS \(\)](#) 函数概述

SYS (2003) - 当前目录

返回默认驱动器或卷上的当前目录或文件夹的名称。

语法

SYS (2003)

返回值类型

字符型

说明

可以使用 SYS (5) 来确认当前驱动器或卷。

请参阅

[SYS \(\) 函数概述](#)

SYS (2004) - Visual FoxPro 启动目录或文件夹

返回启动 Visual FoxPro 的目录或文件夹名称。

语法

SYS (2004)

返回值类型

字符型

说明

在分布式运行时，SYS (2004) 返回 Vfp6r.dll 文件的位置，该文件是使用 Visual FoxPro 专业级版本创建的 Visual FoxPro 应用程序。

示例

```
? 'Visual FoxPro launch directory: ', SYS ( 2004)
```

请参阅

[HOME \(\)](#) , [SET PATH](#) , [SYS \(\)](#) [函数概述](#)

SYS (2005) - 当前资源文件

返回当前 Visual FoxPro 资源文件的名称。

语法

SYS (2005)

返回值类型

字符型

说明

Visual FoxPro 资源文件是一个包含有关系统和用户自定义资源信息的 Visual FoxPro 表，这些信息包括键盘宏、参数选择、系统窗口的位置和大小、日记输入等等。

Visual FoxPro 资源文件默认名是 FOXUSER.DBR，可使用 SET RESOURCE 命令指定一个不同的 Visual FoxPro 资源文件。

示例

```
? 'Current resource file: ', SYS ( 2005)
```

请参阅

[SET RESOURCE, SYS \(\) 函数概述](#)

SYS (2006) - 当前图形适配卡

返回所使用的图形适配卡和显示器的类型。

语法

```
SYS ( 2006 )
```

返回值类型

字符型

示例

```
? 'Current graphics card/monitor: ', SYS ( 2006)
```

请参阅

[ISCOLOR \(\)](#) , [SET DISPLAY, SYS \(\)](#) [函数概述](#)

SYS (2007) - 检查求和值

返回一个字符表达式的检查求和值。

语法

```
SYS ( 2007, cExpression)
```

返回值类型

字符型

参数描述

cExpression

指定字符表达式，SYS (2007) 返回该表达式的检查求和值。

说明

一个检查求和值可用于测试数据的合法性或比较两个字符串。

请参阅

SYS (2010) - CONFIG.SYS 文件设置

返回 CONFIG.SYS 文件中的设置。

语法

SYS (2010)

返回值类型

字符型

说明

在 FoxPro for MS-DOS 中，SYS (2010) 将 CONFIG.SYS 配置文件中的文件设置项作为字符串返回。在 Visual FoxPro 中，SYS (2010) 总是返回 255。。

大多数的 CONFIG.SYS 文件都包含一行语句，指定在 MS-DOS 下可同时打开的最大文件数目。这一行通常为 FILES=NNN，这里 NNN 是一个数，SYS (2010) 返回这个数。

CONFIG.SYS 配置文件中可以没有文件设置项，也可以没有 CONFIG.SYS 文件，对以上两种情况，SYS (2010) 返回默认的 MS-DOS 文件设置值。

SYS (2010) 返回的数目并不是用户在 Visual FoxPro，FoxPro for Windows 和 FoxPro

for MS-DOS 中可以打开的文件数目。MS-DOS 为了满足自身的使用而打开文件，Visual FoxPro 和 FoxPro for Windows 也为其自己内部的使用而打开文件，这些打开文件的数目可能在 FoxPro 的工作期内变化。在 CONFIG.SYS 文件中，文件配置必须略大于要在 Visual FoxPro 中打开的文件数目。

有关 Config.sys 配置文件的详细内容，请参阅 MS-DOS 手册。

请参阅

[SYS \(\) 函数概述](#)

SYS (2011) - 当前锁定状态

返回当前工作区中记录锁定或表锁定的状态。

语法

SYS (2011)

返回值类型

字符型

说明

与 FLOCK () ， LOCK () 和 RLOCK () 函数不同的是，SYS (2011) 不会试图锁定表或记录。

SYS (2011) 返回的字符串与状态栏中显示的信息 ("独占" ， "记录未锁定" ， "记录锁定" …) 相同。

只有在以独占方式打开表的工作站上，SYS (2011) 才返回“独占”；在使用记录锁的工作站上，SYS (2011) 返回“记录锁定”。

请参阅

[FLOCK \(\)](#) , [LOCK \(\)](#) , [RLOCK \(\)](#) , [SYS \(\)](#) 函数概述

SYS (2012) - 备注字段块大小

返回表的备注字段块大小。

语法

SYS (2012 [*nWorkArea* | *cTableAlias*])

返回值类型

字符型

参数描述

nWorkArea | *cTableAlias*

指定表的工作区编号或表的别名，SYS (2012) 返回该表的备注字段块大小。*nWorkArea* 指定工作区编号，*cTableAlias* 指定表的别名。如果不指定工作区或别名，SYS (2012) 返回在当前所选工作区中打开的表的备注字段块大小。如果在指定的工作区中没有打开表或表没有备注字段，那么 SYS (2012) 返回 0。

说明

关于指定表备注字段块大小的详细内容，请参阅 `SET BLOCKSIZE`。

请参阅

[SET BLOCKSIZE](#)

SYS (2013) - 系统菜单名称字符串

返回以空格分隔的字符串，该字符串包含 Visual FoxPro 菜单的内部名称。

语法

`SYS (2013)`

返回值类型

字符型

说明

`SYS (2013)` 方便了使用 Visual FoxPro 系统菜单栏和系统菜单的工作。所返回的字符串包含系统菜单栏的名称、菜单标题和菜单内每项的名称。`SYS (2013)` 可以返回 Visual FoxPro 6.0 的菜单标题名称和菜单项名称。请参阅稍后的“[System Menu Names](#)”。

请参阅

DEFINE BAR, DEFINE MENU, DEFINE POPUP, SYS () 函数概述,
SYS (1500), System Menu Names

SYS (2014) - 最小化路径

返回指定文件相对于当前目录、指定目录或文件夹的最小化路径。

语法

SYS (2014, *cFileName* [, *Path*])

返回值类型

字符型

参数描述

cFileName

指定文件名，SYS (2014) 返回该文件的最小化路径。如果省略 *Path* 参数，则 SYS (2014) 返回该文件与当前目录或文件夹之间的最小路径。可以使用 SET DEFAULT 改变当前目录或文件夹。

Path

指定一个目录或文件夹，SYS (2014) 返回该指定目录或文件夹与 *cFileName* 指定的文件之间的最小化路径。

说明

`SYS (2014)` 和 `FULLPATH ()` 一起使用，使应用程序成为便携式。也就是说，当给定当前程序的位置和该程序所使用的其他文件位置之后，`FULLPATH ()` 和 `SYS (2014)` 就可获得这些文件的正确路径。一旦文件的路径确定了，应用程序就可以访问这些文件。

请参阅

[FULLPATH \(\)](#), [SET DEFAULT](#), [SET PATH](#), [SYS \(\)](#) 函数概述

SYS (2015) - 唯一过程名

返回一个 10 个字符的唯一过程名，该过程名以下划线开头，后接字母和数字的组合。

语法

`SYS (2015)`

返回值类型

字符型

说明

`SYS (2015)` 可用于创建唯一的进程名或函数名。

`SYS (2015)` 返回的名称以系统日期和系统时间来创建。在同一个毫秒间隔中多次调

用 SYS (2015) 将返回唯一的字符串。

请参阅

[SYS \(\) 函数概述, SYS \(3\)](#)

SYS (2016) - SHOW GETS WINDOW 名称

返回最近的 SHOW GETS WINDOW 命令中的窗口名称。包含该函数是为了提供向后兼容性。可用 Refresh 方法来代替。

SYS (2017) - 显示启动屏幕

在 FoxPro 的早期版本中，该函数清除 FoxPro 主窗口，并显示 FoxPro 启动屏幕。包含该函数是为了提供向后兼容性。

SYS (2018) - 错误信息参数

返回最近错误的错误信息参数。

语法

SYS (2018)

返回值类型

字符型

说明

某错误信息返回关于错误原因的附加内容。例如，如果引用一个不存在的变量，该变量的名称就包含在错误信息中，SYS (2018) 返回这一称作错误信息参数的附加内容。变量名和文件名是错误信息参数的一般类型。假设要运行一个名为 REPORTS 的程序，但是该程序并不存在，那么错误信息的文本显示为：

REPORTS 文件不存在。

REPORTS 就是错误信息参数，它可由 SYS (2018) 返回。

? SYS (2018)

请参阅

[ERROR \(\)](#), [MESSAGE \(\)](#), [ON ERROR](#), [SYS \(\)](#) 函数概述

SYS (2019) - 配置文件名称和位置

返回 Visual FoxPro 配置文件的文件名和位置。

语法

SYS (2019)

返回值类型

字符型

说明

Visual FoxPro 配置文件为 Config.fpw，而且该文件在默认情况下没有安装。使用 MODIFY FILE 可以创建自己的 Config.fpw 配置文件。如果不能定位配置文件，SYS (2019) 返回一个空串。Visual FoxPro 配置文件一般位于 Visual FoxPro 启动的目录中，但是也可以放到别的地方。Visual FoxPro 首先在启动目录中寻找该文件。如果配置文件不在启动目录中，则接着搜索 MS-DOS 的路径。有两种方法可以告诉 FoxPro 配置文件的位置及其名称：

- 可以用 MS-DOS 的 SET 命令创建一个名为 FOXPROWCFG 的 MS-DOS 环境变量，然后将配置文件的位置和文件名存入 FOXPROWCFG 中。在 FoxPro for MS-DOS 中，MS-DOS 环境变量名为 FOXPROCFG。
- 也可在启动 Visual FoxPro 时，使用 -C 开关来指定一个配置文件。在 -C 后立

即加上配置文件的位置及文件名。

有关 Visual FoxPro 配置文件的详细内容，请参阅帮助中《安装指南》的第三章 配置 Visual FoxPro”。

请参阅

[MODIFY FILE, SYS \(\) 函数概述](#)

SYS (2020) - 默认磁盘空间

以字节数返回默认磁盘空间。

语法

SYS (2020)

返回值类型

字符型

说明

可以用 SET DEFAULT 命令指定默认磁盘。

请参阅

[DISKSPACE \(\) , SET DEFAULT, SYS \(\) 函数概述](#)

SYS (2021) - 筛选表达式

返回打开的单项索引文件 (.IDX) 的筛选表达式或复合索引文件 (.CDX) 中标识的筛选表达式。

语法

```
SYS (2021, nIndexNumber [, nWorkArea | cTableAlias])
```

返回值类型

字符型

参数描述

nIndexNumber

数值表达式 *nIndexNumber* 指定从打开的索引文件中返回哪一个筛选表达式。当 *nIndexNumber* 从 1 增加到打开的单项索引文件、结构复合索引以及独立的复合索引标识的总数时，SYS (2021) 以下面的顺序从打开的索引文件中返回筛选表达式：

1. 最先从单项索引文件中（如果打开）返回筛选表达式。包含在 USE 或 SET INDEX 命令中单项索引文件的顺序决定返回筛选表达式的顺序。
2. 其次返回在结构复合索引文件（如果存在）中每个标识的筛选表达式。筛选表达式以结构复合索引中标识的创建顺序从标识中返回。

3. 最后返回在任何独立的复合索引文件中每个标识的筛选表达式。筛选表达式以独立的复合索引中标识的创建顺序从标识中返回。

如果 *nIndexNumber* 大于打开的单项索引文件、结构索引文件标识以及独立的复合索引文件标识的总数时，返回一个空串。

nWorkArea

为另一个工作区中打开的表指定工作区编号。

cTableAlias

为另一个工作区中打开的表指定一个别名。如果表没有指定的别名，Visual FoxPro 将显示一条错误信息。

说明

可以在 Visual FoxPro 中创建已筛选的索引。如果在 INDEX 中包含可选的 FOR 子句，那么索引文件作为一个筛选条件作用在表上。在 FOR 子句只有满足筛选表达式 *lexpression* 的记录可以显示和访问。在索引文件中，为那些满足筛选表达式的记录创建索引关键字。

如果没有用 FOR 子句创建索引或索引标识，则返回一个空串。

USE 和 SET INDEX 命令都支持索引文件列表，该列表可以为一个表打开多个索引文件。在索引文件列表中可以包含任何单项索引文件名、结构复合索引文件和独立的复合索引文件名的组合。

除非指定特殊的工作区或别名，否则 SYS(2021) 从在当前工作区中打开的索引文件中返回筛选表达式。

请参阅

CDX () , KEY () , INDEX , MDX () , NDX () , REINDEX , SET

[INDEX,SET ORDER,](#)
[SYS \(\) 函数概述, TAG \(\)](#)

SYS (2022) - 磁盘簇 (块) 大小

以字节为单位返回指定磁盘簇 (块) 的大小。

语法

`SYS (2022 [, cDiskName]`)

返回值类型

字符型

参数描述

`cDiskName`

指定磁盘驱动器字母 (A, B, C ...), 而不是当前默认磁盘。

请参阅

[SET DEFAULT, SYS \(\) 函数概述](#)

SYS (2023) -临时文件驱动器

返回 Visual FoxPro 存储临时文件的驱动器和目录。

语法

SYS (2023)

返回值类型

字符型

说明

可以在 Visual FoxPro 配置文件中包含特殊的 TMPFILES 配置项，指定存储临时文件的驱动器和目录。

有关特殊的 TMPFILES 配置项的详细内容，请参阅帮助中《安装指南》的第三章“配置 Visual FoxPro”。

请参阅

[SYS \(\) 函数概述](#)

SYS (2029) -表类型

返回与表类型对应的值。

语法

SYS (2029 [, *nWorkArea* | *cTableAlias*])

返回值类型

字符型

参数描述

nWorkArea

指定打开表的工作区。

cTableAlias

指定表的别名。如果省略 *nWorkArea* 和 *cTableAlias* 参数，SYS (2029) 将返回在当前工作区中打开表的类型值。

说明

下表列出了 SYS (2029) 的返回值类型及其对应的表类型。

返回值类型

表类型

0	没有表打开
3	早期的 FoxPro, FoxBASE+, dBASEIII PLUS 和 dBASE IV 版本, 没有备注字段。
48	有或没有备注字段的 Visual FoxPro 表
67	没有备注字段的 dBASE IV SQL 表
99	具有备注字段的 dBASE IV 系统表
131	具有备注字段的 FoxBASE+ 和 dBASEIII PLUS 表
139	具有备注字段的 dBASE IV 表
203	具有备注字段的 dBASE IV SQL 表
245	具有备注字段的早期 FoxPro 版本

示例

下面的示例打开 testdata 数据库中的 testdata 表。

```
CLOSE DATABASES  
OPEN DATABASE (HOME (2) + 'data\testdata')  
USE Customer && 打开 "customer" 表
```

```
CLEAR  
DO CASE  
    CASE SYS (2029) = '3'  
        ? 'Previous versions of FoxPro'  
    CASE SYS (2029) = '48'
```

```
    ? 'Visual FoxPro Table'
CASE SYS ( 2029 ) = '67'
    ? 'dBASE IV SQL table, no memo fields'
CASE SYS ( 2029 ) = '99'
    ? 'dBASE IV SQL System table with a memo field'
CASE SYS ( 2029 ) = '131'
    ? 'FoxBASE+ table with a memo field'
CASE SYS ( 2029 ) = '139'
    ? 'dBASE IV table with a memo field'
CASE SYS ( 2029 ) = '203'
    ? 'dBASE IV SQL table with a memo field'
CASE SYS ( 2029 ) = '245'
    ? 'Previous versions of FoxPro with a memo field'
ENDCASE
```

请参阅

[ALTER TABLE -SQL, CREATE, CREATE TABLE -SQL, USE](#)

SYS (2333) - 开启或关闭 ActiveX 的双界面支持

允许或禁止 ActiveX 的双界面支持 (VTABLE binding)。

语法

SYS (2333 [, 0 | 1 | 2])

返回值类型

字符型

参数描述

0

禁止 ActiveX 的双界面支持。如果省略 0 或 1，则禁止 ActiveX 的双界面支持。0 是 Visual FoxPro 6.0 的默认设置。

1

允许 ActiveX 的双界面支持。1 是 Visual FoxPro 5.0 的默认设置。

2

返回 SYS (2333) 的当前设置 (0 或 1) 。

说明

ActiveX 双界面 (VTABLE binding) 是 Visual FoxPro 支持的一种优化功能。如果 ActiveX 本身没有使用双界面，您可以禁止 Visual FoxPro 实施这种优化。

如果 ActiveX 控件在创建一个实例时不能正常工作，请在 ActiveX 控件创建实例前使用 SYS (2333) 或 SYS (2333, 0) 禁止双界面支持。在实例创建成功后，再使用 SYS (2333, 1) 开启双界面支持。

请参阅

[ActiveX 控件概览](#), [OLE Container 控件](#)



返回总目录

- SYS (2334)** — 自动服务启动
- SYS (2335)** — 无人照管服务模式
- SYS (3004)** — 返回环境 ID 值
- SYS (3005)** — 设置环境 ID 值
- SYS (3006)** — 设置语言和环境 ID 值
- SYS (3050)** — 设置缓冲内存大小
- SYS (3051)** — 设定锁定重试间隔
- SYS (3052)** — 忽略 SET REPROCESS 锁定
- SYS (3053)** — ODBC 环境句柄
- SYS (3054)** — 显示 Rushmore 优化的级别
- SYS (3055)** — FOR and WHERE 子句复杂级别
- SYS (3056)** — 读取注册设置
- SYS (4204)** — Active Document 调试
- SYSMETRIC () 函数**
- 系统变量概述**
- System Menu Names**
- TabIndex 属性**
- TABLEREVERT () 函数**
- TABLEUPDATE () 函数**

Tabs 属性

_TABS 系统变量

TabStop 属性

TabStretch 属性

TabStyle 属性

Tag 属性

TAG () 函数

TAGCOUNT () 函数

TAGNO () 函数

_TALLY 系统变量

TAN () 函数

TARGET () 函数

TerminateRead 属性

Text 属性

TEXT ... ENDTEXT 命令

_TEXT 系统变量

TextBox 控件

TextHeight 方法

TextWidth 方法

THIS 对象

THISFORM 对象

THISFORMSET Object Reference

_THROTTLE 系统变量

TIME () 函数

Timer 控件

Timer 事件

TitleBar 属性

ToolBar 对象

ToolTipText 属性

Top 属性

TopIndex 属性

TopItemID 属性

TOTAL 命令

TRANSFORM () 函数

_TRANSPORT 系统变量

_TRIGGERLEVEL 系统变量

SYS (2334) — 自动服务启动

返回一个值，表明如何激活一个 Visual FoxPro 自动服务程序，或者是否运行一个独立的可执行文件（.exe）。

语法

SYS (2334)

返回值类型

字符型

说明

下表列出了 SYS (2334) 为自动服务程序的返回值类型

返回值类型

激活方法

0	未知（例如，从 INIT 方法中激活，既不是 VTable 也不是 Idispatch）
1	通过 Vtable 绑定的 OLEPUBLIC 方法激活。
2	IDispatch

当从一个独立的可执行文件（.exe）中执行时，SYS (2334) 也返回零。使用 StartMode 属性可以判断 Visual FoxPro 的一个实例是如何启动的。

请参阅

[StartMode 属性](#), [SYS \(\) 函数概述](#)

SYS (2335) –无人照管服务模式

启用或废止可发布的 Visual FoxPro .exe 自动服务程序的模式状态。

语法

SYS (2335 [, 0 | 1])

返回值类型

字符型

参数描述

0

启用无人照管模式。当启用无人照管模式时，每当发生一个模式状态时，都会产生一个 Visual FoxPro 错误。使用 ON ERROR 例程，.exe 自动服务程序可以捕捉这些错误。

1

废止无人照管模式。会发生需要用户干预的模式状态。在启动时废止无人照管模式。

说明

使用 SYS (2335) 可以启用或废止 Visual FoxPro .exe 自动服务程序的模式状态。自动服务程序是使用项目管理器创建的。

当显示对话框或错误信息时，会发生模式状态，要退出这个对话框或错误信息，并且

继续执行程序，需要用户的输入。在远程启动的 .exe 服务程序可能不希望模式状态，也不需要用户的干预。在模式状态下，会停止程序的执行，并且为了让程序继续执行，需要干预。

下表列出了 .exe 服务程序可能发生的几种典型的模式状态示例。

模式状态

示例

WAIT 命令或 MESSAGEBOX () 函数

会在程序代码中发生。

Visual FoxPro 错误，例如“对文件的访问被拒绝”或“超过了允许的 DO 嵌套级别”

会在程序代码中发生。

“打开”对话框

当一个 SQL 语句中包含的表不能加载时发生。

“SQL 连接登录”对话框

当不能建立连接时发生。

在无人照管的 .exe 自动服务程序代码中，SYS (2335 ,0) 应该尽早执行，因为在程序运行后，随时都可能进入模式状态。

注意 SYS (2335) 只应用于 StartMode 属性等于 2 的 .exe 自动服务程序。对于内部处理 .dll 自动服务程序 (它的 StartMode 属性等于 3) ，总是启用无人照管模式。在运行时刻应用程序中不带参数地使用 SYS (2335) ，会返回它的当前设置。

请参阅

[ON ERROR, StartMode 属性, SYS \(\) 函数概述](#)

SYS (3004) – 返回环境 ID 值

返回自动化和 ActiveX 控件使用的环境 ID 值。

语法

SYS (3004)

返回值类型

字符型

说明

SYS (3004) 返回当前 Visual FoxPro 的环境 ID 值 (LCID)，该环境 ID 值由 Visual FoxPro 语言 ID 值 (LangID) 和排序 ID 值决定。

环境 ID 值决定 OLE 自动化和 OLE 控件交换信息时所用的语言。默认的 Visual FoxPro 环境 ID 值是 1033 — 使用英语。

例如，假设已经安装了 Microsoft Excel 5.0 的德语版本，该版本支持英语和德语。在这样的情况下，下面的示例可以启动和关闭 Microsoft Excel 5.0 的德语版本。

```
oleExcel1 = CREATEobject ('Excel.Application')  && 启动 Excel
```

```
? SYS (3005, 1033)  && 英语环境 ID 值
```

```
oleExcel.Quit  && 用英语命令关闭 Excel
```

```
oleExcel2 = CREATEobject ('Excel.Application')  && 启动 xcel
```

```
? SYS (3005, 1031)  && 德国环境 ID 值
```

oleExcel.Beenden && 用德语命令关闭 Excel

关于 Visual FoxPro 环境 ID 值的列表，请参阅 SYS (3005) - 设置环境 ID 值。
关于 Locale、Language 和 Sort Ids 的详细内容，请在 MSDN 中搜索。
请参阅

[DefOLELCID 属性](#)，[OLELCID 属性](#)，[SYS \(3005\)](#)，[SYS \(3006\)](#)

SYS (3005) – 设置环境 ID 值

设置自动化和 ActiveX 控件使用的环境 ID 值。

语法

SYS (3005, *nLocaleID*)

返回值类型

字符型

参数描述

nLocaleID

指定环境 ID 值，下表包含了 Visual FoxPro 支持的环境 ID 值：

nLocaleID

语言

1029	捷克语
1031	德语
1033	英语（默认）
1034	西班牙语
1036	法语
1040	意大利语
1045	波兰语
1046	葡萄牙语（巴西）
2070	葡萄牙语（标准）

说明

SYS（3005）设置全球的环境 ID 值。环境 ID 值决定 OLE 自动化和 OLE 控件交换信息时所使用的语言。默认的 Visual FoxPro 环境 ID 值是 1033 — 使用英语。

注意 设置表单或 Visual FoxPro 主窗口的环境 ID 值最适当的方法是使用 DefOLELCID 属性。OLE 自动化命令语言仅受由 SYS（3005）设置的全球环境 ID 值的影响。DefOLELCID 属性和 OLELCID 属性仅影响显示 OLE 控件的用户接口语言，而不是 OLE 自动化命令语言。

例如，假设已经安装了 Microsoft Excel 5.0 的德语版本（该版本支持英语和德语），在

这样的情况下，下面的示例可以启动和关闭 Microsoft Excel 5.0 的德语版本。

```
oleExcel1 = CREATEobject ('Excel.Application')  && 启动 Excel  
? SYS ( 3005, 1033 )  && 英国环境 ID 值  
oleExcel.Quit  && 用英语命令关闭 Excel  
oleExcel2 = CREATEobject ('Excel.Application')  && 启动 Excel  
? SYS ( 3005, 1031 )  && ID 德国环境 ID 值  
oleExcel.Beenden  && 用德语命令关闭 Excel
```

关于 Locale IDs 的详细内容，请在 MSDN 中搜索。

请参阅

[DefOLELCID 属性](#), [OLELCID 属性](#), [SYS \(3004 \)](#), [SYS \(3006 \)](#)

SYS (3006) – 设置语言和环境 ID 值

设置语言 ID 值和环境 ID 值。

语法

```
SYS ( 3006, nLanguageID )
```

返回值类型

字符型

参数描述

nLanguageID

指定语言 ID 值。

说明

SYS (3006) 设置 Visual FoxPro 语言 ID 值 (LangID)，在语言 ID 值和当前排序 ID 值的基础上设置环境 ID 值。

注意 设置表单或 Visual FoxPro 主窗口的环境 ID 值最适当的方法是使用 DefOLELCID 属性。

关于 Language、Locale 和 Sort Ids 的详细内容，请在 MSDN 中搜索。

请参阅

[DefOLELCID 属性](#)，[OLELCID 属性](#)，[SYS \(3004\)](#)，[SYS \(3005\)](#)

SYS (3050) – 设置缓冲内存大小

设置前台或后台缓冲内存大小。

语法

SYS (3050, *nType*, [*nBuffMemSize*])

返回值类型

字符型

参数描述

nType

指定缓冲区，下表列出了 *nType* 的值及其对应的缓冲区。

nType	缓冲区
1	前台缓冲区
2	后台缓冲区

nBuffMemSize

指定以字节计算的最大缓冲内存的大小。如果指定了一个小于 256 K 字节的 *nBuffMemSize* 值，Visual FoxPro 将设置缓冲内存大小为 256 K 字节。将 *nBuffMemSize* 设为 0，可以把缓冲内存大小重置为 Visual FoxPro 启动时的数值，该数值依赖于所用的计算机内存的大小。如果省略 *nBuffMemSize* 参数，SYS (3050) 返回缓冲内存大小。

说明

通过调整 Visual FoxPro 分配给前台和后台缓冲区的内存大小，SYS (3050) 可以优化 Visual FoxPro 的性能。

前台内存缓冲区是当 Visual FoxPro 在前台执行当前活动应用程序时的可用内存，后台内存缓冲区是 Visual FoxPro 在后台执行另一个后台应用程序时的可用内存。

SYS (3050) 将一个数值作为字符串返回，该字符串表示 Visual FoxPro 分配给前台或后台缓冲区的最大内存。

请参阅

[SYS \(1001\)](#)

SYS (3051) — 设定锁定重试间隔

指定在一次锁定尝试失败之后，再次尝试锁定记录表、备注或索引文件之前，Visual FoxPro 等待的毫秒时间值。

语法

SYS (3051, [*nWaitMilliseconds*])

返回值类型

字符型

参数描述

nWaitMilliseconds

指定以毫秒为单位的等待时间，可以是一个从 100 到 1000 毫秒之间的值。指定 *nWaitMilliseconds* 为 0，可以返回 Visual FoxPro 启动时默认的锁定尝试时间间隔数值（333 毫秒）。如果省略了 *nWaitMilliseconds* 参数，SYS (3051) 返回当前的锁定尝试时间间隔。

说明

SYS (3051) 将一个数值作为字符串返回，该字符串表示锁定尝试时间间隔。

请参阅

[SET REPROCESS, SYS \(3052\)](#)

SYS (3052) – 忽略 SET REPROCESS 锁定

指定当尝试锁定一个索引或备注文件时， Visual FoxPro 是否使用 SET REPROCESS 设置。

语法

SYS (3052, *nFileType*, [*lHonorReprocess*])

返回值类型

字符型

参数描述

nFileType

指定文件类型，下表列出了 *nType* 的值及其对应的文件类型：

<i>nFileType</i>	文件类型
------------------	------

1	索引文件
2	备注文件

lHonorReprocess

指定当对索引或备注文件的锁定尝试失败时， Visual FoxPro 是否使用 SET REPROCESS 设置。

当 Visual FoxPro 尝试锁定用 *nFileType* 指定的文件时，指定 *lHonorReprocess* 为“真” (.T.)，将使用 SET REPROCESS 设置；指定 *lHonorReprocess* 为

“假” (.F.) (默认值)，将忽略 SET REPROCESS 设置。当设置为假时，Visual FoxPro 将无限期等待锁定指定文件，这和 FoxPro 早期版本中的锁定操作相同。

如果省略了 *lHonorReprocess* 参数，SYS (3052) 返回用 *nFileType* 指定的文件类型的当前设置。

说明

在 Visual FoxPro 中，SYS (3052) 提供文件锁定上的附加控制。如果您的应用程序使用事务处理，那么最好将 *lHonorReprocess* 设置为“真” (.T.)，以减少文件锁定冲突的危险。SYS (3052) 将数值 0 — “假” (.F.) 或 1 — “真” (.T.) 作为一个字符串返回。如果 SYS (3052) 中包含 *lHonorReprocess* 参数，返回值类型与 *lHonorReprocess* 指定的逻辑值相同。如果省略了 *lHonorReprocess* 参数，返回值类型是用 *nType* 指定的文件类型的当前设置。

请参阅

[SET REPROCESS, SYS \(3051\)](#)

SYS (3053) — ODBC 环境句柄

返回 ODBC 环境句柄。

语法

SYS (3053)

返回值类型

字符型

说明

如果没有加载 ODBC，SYS (3053) 就加载 ODBC 并返回 ODBC 环境句柄。SYS (3053) 返回的环境句柄通过调用 ODBC API 提供对 ODBC 的访问。在 Visual FoxPro 中，通过 DECLARE.DLL 和在 Visual FoxPro 专业版中创建的外部 Visual FoxPro API 库例程可以访问 ODBC API。

注意，某一时刻只能使用一个 ODBC 环境句柄，所以使用 ODBC 调用的 Visual FoxPro 应用程序应该使用 SYS (3053) 获得 ODBC 环境句柄，而不是通过 ODBC API 调用来再分配或释放 ODBC 环境句柄。操作 ODBC 环境句柄时应当谨慎。

关于 ODBC API 的详细内容，请在 MSDN 中搜索。

请参阅

[DECLARE - DLL, SET LIBRARY](#)

SYS (3054) – 显示 Rushmore 优化的级别

允许或禁止显示查询的 Rushmore 优化级别。

语法

`SYS (3054, 0 | 1 | 11)`

返回值类型

字符型

参数描述

0

(默认) 禁止显示 Rushmore 优化级别。

1

允许显示 Rushmore 筛选优化的级别。

11

允许显示 Rushmore 连接优化的级别。

说明

`SYS (3054)` 将参数 (0, 1 或 11) 返回为一个字符串。

您可以使用 `SYS (3054)` 查看一个查询在何种程度上使用了 Rushmore 技术。

运行 `SYS (3054, 1)`，当查询执行完之后，筛选优化的级别就显示在当前活动的窗口中。

`SYS (3054, 1)` 显示 Rushmore 筛选优化的三个级别和可被优化的索引标识。如下表：

优化的级别

说明

无

查询不能使用 Rushmore 优化。

续表

部分	查询中的某些表达式可被优化。同时列出所有可被 Rushmore 优化的索引标识。
全部	查询可被完全优化。同时列出被 Rushmore 优化的索引标识。

您通过 SYS (3054,1) 检测到某个查询未被优化或只是部分优化后，可以修改这个查询，使之充分利用 Rushmore 技术。

使用 SYS (3054, 11)，可以在一次查询之后在激活的窗口中显示 Rushmore 优化级别。

使用 SYS (3054, 0)，可以在一次查询之后停止显示 Rushmore 优化级别。

有关 Rushmore 技术和优化查询的详细内容，请参阅《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”。

请参阅

[SELECT—SQL 命令](#)

SYS (3055) — FOR and WHERE 子句复杂级别

在支持 FOR 和 WHERE 子句的命令和函数中设置它们的复杂级别。

语法

SYS (3055 [, nComplexity])

返回值类型

字符型

参数描述

nComplexity

设置复杂级别。NComplexity的有效范围是从320到2040。默认值是320。如果指定了一个奇数，则四舍五入到最接近的偶数。

说明

如果得到错误1308——堆栈空间不足或1812——SQL: 语句太长，可以增加FOR和WHERE子句的复杂程度来避免此错误。

例如，对于不使用关键字的本地表或视图，调用TABLEUPDATE ()会产生一个长WHERE子句，以查找更新的行。WHERE子句中支持的字段默认数是40。如果得到错误1812——SQL: 语句太长，可以为更新使用关键字，或者用SYS (3055) 来增加WHERE子句的复杂程度。如果使用SYS (3055) ，其值增加到表中字段数量的8倍。

= SYS (3055, 8 * MIN (40, FCOUNT ()))

如果发出SYS (3055) 时没有nComplexity参数，则返回当前的设置。

下面的命令和函数支持FOR和WHERE子句：

APPEND FROM

APPEND FROM

AVERAGE

ARRAY

BLANK

BROWSE

CALCULATE

CHANGE	COPY TO ARRAY	COPY TO
COUNT	DEFINE PAD	DELETE
DELETE - SQL	DISPLAY	EXPORT
FOR ()	INDEX	LABEL
LIST	LOCATE	RECALL
REPLACE	REPLACE FROM	REPORT
	ARRAY	
SCAN ...	SELECT - SQL	SORT
ENDSCAN		
SUM	TABLEUPDATE	UPDATE - SQL
	()	

请参阅

TABLEUPDATE () , SYS () 函数概述

SYS (3056) — 读取注册设置

让 Visual FoxPro 再次读取自己的注册表设置，并且使用当前的注册表设置更新自己。

语法

SYS (3056 [,1])

返回值类型

字符型

参数描述

1

包含 1 选项可以只根据注册表设置进行更新，不包括 SET 命令和文件位置。

说明

返回空字符串。

Visual FoxPro 在 Windows 注册表的选项对话框中保存设置。当 Visual FoxPro 启动时，读取这些设置，并利用这些设置进行配置。SYS (3056) 可更新包含有文件名和路径的系统变量，而系统变量由 SET 命令和注册表设置。

当您选择 Set as a default 时，选项对话框指定的设置写入注册表。其中一些影响 Visual FoxPro 环境的设置含有 SET 命令，例如 SET BELL，SET LOCK 等。注册表设置也可以通过 Windows API 来改变，请参阅位于 Visual Studio...\Samples\Vfp98\Solution 目录的 Solution.app 例子。

注意 SYS (3056) 不更新来自 Config.fpw (即 Visual FoxPro 配置文件) 的设置。

请参阅

选项对话框

SYS (4204) — Active Document 调试

在 Visual FoxPro 的调试程序中，启用或废止对 Active Documents 的调试支持。

语法

SYS (4204 [, 0 | 1])

返回值类型

字符型

参数描述

0

在 Visual FoxPro 中关闭对 Active Document 的包含，使得不能在调试程序中调试一个 Active Document。

1

(默认值) 在 Visual FoxPro 中打开对 Active Document 的调试支持，使得可以在调试程序中调试一个 Active Document。

说明

返回空字符串。

使用 SYS (4204) 可以在 Visual FoxPro 的调试程序中调试 Active Documents。执行 SYS (4204, 1)，就让 Visual FoxPro 成为一个 Active Document 服务程序，就可以调试一个 Active Document 了。执行 SYS (4204, 0) 可以关闭对 Active Document 的包

含。

注意 为了打开调试程序，必须在 Active Document 代码中添加 SET STEP ON。
请参阅

[_RUNACTIVEDOC 系统变量](#)

SYSMETRIC () 函数

返回操作系统屏幕元素的大小。

语法

SYSMETRIC (*nScreenElement*)

返回值类型

数值型

参数描述

nScreenElement

指定屏幕元素。下表显示了 *nScreenElement* 的值及其对应的屏幕元素：

NScreen Element	屏幕元素
----------------------------	-------------

1	屏幕宽度
---	------

2	屏幕高度
3	可调整窗口的边框宽度
4	可调整窗口的边框高度
5	垂直滚动栏上滚动箭头的宽度
6	垂直滚动栏上滚动箭头的高度
7	水平滚动栏上滚动箭头的宽度
8	水平滚动栏上滚动箭头的高度
9	窗口标题的高度
10	不可调整的窗口边框的宽度
11	不可调整的窗口边框的高度
12	DOUBLE 或 PANEL 窗口边框的宽度
13	DOUBLE 或 PANEL 窗口边框的高度
14	文本编辑窗口中水平滚动栏上滚动框的宽度
15	文本编辑窗口中水平滚动栏上滚动框的高度
16	最小化窗口图标的宽度
17	最小化窗口图标的高度
18	最大化插入点宽度
19	最大化插入点高度
20	单线菜单栏高度
21	最大化窗口宽度
22	最大化窗口高度
23	"Kanji" 窗口高度
24	最小化可调整窗口的宽度
25	最小化可调整窗口的高度

26	最小化窗口宽度
27	最小化窗口高度
28	窗口控件的宽度
29	窗口控件的高度
30	如果有鼠标，为 1；否则为 0。
31	对于 Microsoft Windows 测试版本为 1，否则为 0
32	鼠标按钮交换了为 1，否则为 0。
33	在半标题窗口的标题或标题栏中一个按钮的宽度。
34	半标题窗口标题区的高度。

说明

SYSMETRIC () 返回屏幕元素的大小。屏幕元素包括菜单、窗口、窗口控件和插入点。除非另行注明，否则以像素为单位返回一个数值。该返回值类型可能随不同的显示器、显示驱动程序和视频硬件而变化。有关 Microsoft Windows 屏幕元素更深入的内容，请参阅《Microsoft Windows 中文版程序员指南》中的 GetSystemMetrics 函数。

SYSMETRIC () 在 Visual FoxPro 中，可以确定创建的菜单、窗口和窗口控件的大小。用 DEFINE WINDOWS 和 DEFINE MENU 创建的窗口、菜单使用与操作系统的窗口、菜单相同的屏幕元素尺寸。

请参阅

FONTMETRIC () , GETFONT () , TXTWIDTH () , WFONT ()

系统变量概述

系统变量是 Visual FoxPro 自动创建并维护的内置变量。默认情况下，它们的属性是 PUBLIC，也可以声明成 PRIVATE 属性。

在 Visual FoxPro 中，许多原有的系统变量功能已经被新的功能代替。有关的详细内容，请参阅每个系统变量的说明。

下表列出了五种 Visual FoxPro 系统变量以及它们在表达式中的指示符：

变量类型	说明	表达式
C	字符型	cExpression
D	日期型	dExpression
L	逻辑值	lExpression
N	数值型	nExpression
O	对象型	oExpression

下表列出了全部系统变量的类型和默认值：

变量	类型	默认值
<code>_ALIGNMENT</code>	C	LEFT
<code>_ASCIICOLS</code>	N	80
<code>_ASCIROWS</code>	N	63

续表

_ASSIST	C	在 Visual FoxPro 中是空串，否则是 CATALOG.APP
_BEAUTIFY	C	在 Visual FoxPro 中是空串，否则是 BEAUTIFY.APP
_BOX	L	.T.
_BROWSER	C	Browser.app
_BUILDER	C	Builder.app
_CALCMEM	N	0.0
_CALCVALUE	N	0.0
_CLIPTEXT	C	空串
_CONVERTER	C	Convert.app
_COVERAGE	C	Coverage.app
_CUROBJ	N	-1
_DBLCLICK	N	0.5
_DIARYDATE	D	当前日期
_DOS	L	在 FoxPro for MS-DOS 中为“真” (.T.)
_FOXDOC	C	在 Visual FoxPro 中是空串，否则是 FOXDOC.EXE
_GALLERY	C	Gallery.app
_GENGRAPH	C	在 Visual FoxPro 中是空串，否则是 GENGRAPH.PRG
_GENHTML	C	Genhtml.prg
_GENMENU	C	Genmenu.prg

续表

_GENPD	C	在 Visual FoxPro 中是空串，否则是 GENPD.APP
_GENSCRN	C	Genscrn.prg
_GENXTAB	C	在 Visual FoxPro 中是空串，否则是 GENXTAB.PRG
_GETEXPR	C	空字符串
_INCLUDE	C	Empty string
_INDENT	N	0
_LMARGIN	N	0
_MAC	L	.T. 对于 Visual FoxPro for Macintosh
_MLINE	N	0
_PADVANCE	C	FORMFEED
_PAGENO	N	1
_PBPAGE	N	1
_PCOLNO	N	当前列
_PCOPIES	N	1
_PDRIVER	C	空串
_PDSETUP	C	空串
_PECODE	C	空串
_PEJECT	C	NONE
_PEPAGE	N	32767
_PLENGTH	N	66
_PLINENO	N	0
_PLOFFSET	N	0

续表

_PPITCH	C	默认值
_PQUALITY	L	.F.
_PRETEXT	C	空串
_PSCODE	C	空串
_PSPACING	N	1
_PWAIT	L	.F.
_RMARGIN	N	80
_RUNACTIVEDOC	C	Runactd.prg
_SCCTEXT	C	Scctext.prg
_SCREEN	O	FORM
_SHELL	C	空串
_SPELLCHK	C	Spellchk.app
_STARTUP	C	在 Visual FoxPro 中是空串，否则是 STARTUP.APP
_TABS	C	空串
_TALLY	N	0
_TEXT	C	-1
_THROTTLE	N	0
_TRANSPORT	C	在 Visual FoxPro 中是空串，否则是 TRANSPORT.PRG
_TRIGGERLEVEL	N	0
_UNIX	L	在 FoxPro for UNIX 中为“真” (.T.)

续表

<code>_VFP</code>	O	Microsoft Visual FoxPro Application 6.0
<code>_WINDOWS</code>	L	在 Visual FoxPro 中为“真” (.T.)
<code>_WIZARD</code>	C	Wizard.app
<code>_WRAP</code>	L	.F.

请参阅

`PUBLIC, PRIVATE, STORE`

System Menu Names

下表列出了 Visual FoxPro 系统菜单栏、菜单标题和菜单项的内部名称和界面名称。利用这些名称，您可以使用 Visual FoxPro 系统菜单栏为您自己的应用程序创建一个菜单系统。

`SYS (2013)` 函数可以返回空格分隔的字符串，其中包含这里的名称。若要查看内部的系统菜单名称，也可以使用“菜单设计器”创建一个快速菜单。

有关系统菜单栏的详细信息，请参阅《*Microsoft Visual FoxPro 6.0 中文版程序员指南*》中的第十一章“设计菜单和工具栏”。

Visual FoxPro 中的系统菜单名称

Visual FoxPro 本身的系统菜单栏的内部名称为 `_MSYSMENU`。

系统菜单

菜单标题	内部名称
文件	<code>_MSM_FILE</code>
编辑	<code>_MSM_EDIT</code>
数据工作期	<code>_MSM_VIEW</code>
格式	<code>_MSM_TEXT</code>
工具	<code>_MSM_TOOLS</code>
程序	<code>_MSM_PROG</code>
窗口	<code>_MSM_WINDO</code>
帮助	<code>_MSM_SYSTM</code>

文件菜单 — Visual FoxPro

菜单标题	内部名称
文件菜单	<code>_MFILE</code>
新建 ...	<code>_MFI_NEW</code>

续表

打开 ...	_MFI_OPEN
关闭	_MFI_CLOSE
全部关闭	_MFI_CLALL
第一个分隔符	_MFI_SP100
保存	_MFI_SAVE
另存为 ...	_MFI_SAVAS
另存为 HTML...	_MFI_SAVASHTML
还原	_MFI_REVRT
第二个分隔符	_MFI_SP200
导入	_MFI_IMPORT
导出	_MFI_EXPORT
第三个分隔符	_MFI_SP300
页面设置 ...	_MFI_PGSET
打印预览	_MFI_PREVU
打印 ...	_MFI_SYSPRINT

续表

发送	_MFI_SEND
第四个分隔符	_MFI_SP400
退出	_MFI_QUIT

编辑菜单 — Visual FoxPro

菜单标题	内部名称
编辑菜单	_MEDIT
撤消	_MED_UNDO
重做	_MED_REDO
第一个分隔符	_MED_SP100
剪切	_MED_CUT
复制	_MED_COPY
粘贴	_MED_PASTE
选择性粘贴	_MED_PSTLK
清除	_MED_CLEAR
第二个分隔符	_MED_SP200
全部选定	_MED_SLCTA
第三个分隔符	_MED_SP300

续表

查找 ...	_MED_FIND
再次查找	_MED_FINDA
替换 ...	_MED_REPL
定位行 ...	_MED_GOTO
第四个分隔符	_MED_SP400
插入对象 ...	_MED_INSOB
对象 ...	_MED_OBJ
链接 ...	_MED_LINK
转换为静态	_MED_CVTST
第五个分隔符	_MED_SP500
属性	_MED_PREF

显示菜单

菜单标题

内部名称

显示菜单
工具栏 ...

_MVIEW
_MVI_TOOLB

工具菜单

菜单标题

内部名称

工具菜单	_MTOOLS
向导	_MTL_WZRDS
第一个分隔符	_MTL_SP100
拼写检查 ...	_MTL_SPELL
宏 ...	_MST_MACRO
类浏览器	_MTL_BROWSER
组件管理库	_MTL_GALLERY
代码范围分析器	_MTL_COVERAGE
修饰 ...	_MED_BEAUT
运行 Active Document...	_MTI_RUNACTIVEDOC
第二个分隔符	_MTL_SP200
调试器	_MTL_DEBUGGER
第三个分隔符	_MTL_SP400
选项 ...	_MTL_OPTNS

程序菜单

菜单标题

内部名称

程序菜单

_MPROG

运行...

_MPR_DO

取消

_MPR_CANCL

继续执行

_MPR_RESUM

挂起

_MPR_SUSPEND

第一个分隔符

_MPR_SP100

编译...

_MPR_COMPL

窗口菜单

菜单标题

内部名称

窗口菜单

_MWINDOW

全部重排

_MWI_ARRAN

第一个分隔符

_MWI_SP100

隐藏

_MWI_HIDE

续表

全部隐藏	_MWI_HIDE
全部显示	_MWI_SHOWA
清除	_MWI_CLEAR
循环	_MWI_ROTAT
第二个分隔符	_MWI_SP200
命令窗口	_MWI_CMD
数据工作期	_MWI_VIEW

帮助菜单

菜单标题

内部名称

帮助菜单	_MSYSTEM
Microsoft Visual FoxPro 帮助主题	_MST_HPSCCH
第一个分隔符	_MST_SP100
目录	_MST_MSDNC
索引	_MST_MSDNI
搜索	_MST_MSDNS
技术支持	_MST_TECHS
Microsoft on the Web	_HELPWEBVFPFREESTUFF

第二个分隔符	_MST_SP200
关于 Microsoft Visual FoxPro...	_MST_ABOUT

请参阅

[SYS \(2013\)](#) , [SYS \(1500\)](#)

TabIndex 属性

指定页面上控件的 Tab 键次序，以及表单集中表单对象的 Tab 键次序。设计和运行时可用。

语法

[Object.]Control.TabIndex[= nOrder]

参数描述

nOrder

指定控件的 Tab 键次序。

说明

对表单中的控件来说，按照控件添加的次序指定 Tab 键次序，每个新建控件都位于 Tab 键次序的末尾。

对于表单集中的一个表单，按照表单添加的次序指定 Tab 键次序。在表单集中，*nOrder* 为 1 的表单就是第一个被激活的表单。当用户在表单的最后一个控件上按下 Tab 键，焦点就移到下个表单中 Tab 键次序为 1 的控件上。如果表单不在表单集中，则忽略 TabIndex 属性。

可以在设计时使用属性窗口或在运行时使用代码改变此属性。然而，如果要改变一个控件或表单的设置，必须改变页面中所有控件和表单集中所有表单的设置值。如果没有指明所有控件或表单对象的 Tab 键次序，对于剩下的控件和表单对象，Visual FoxPro 将把它们运行时添加、改变的次序作为 Tab 键次序，这可能产生意料不到的结果。

若要忽略一个控件或表单的 Tab 键次序，可将 TabStop 属性设置为“假”（.F.）。

应用于

复选框，组合框，命令按钮，命令组，容器对象，控件对象，编辑框，表单，表格，标签，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，_SCREEN，微调，文本框

请参阅

TabStop 属性

TABLEREVERT () 函数

放弃对缓冲行、缓冲表或临时表的修改，并且恢复远程临时表的 OLDVAL () 数据以及本地表和临时表的当前磁盘数值。

语法

```
TABLEREVERT ( [lAllRows] [, cTableAlias | nWorkArea] )
```

返回值类型

数值型

参数描述

lAllRows

决定是否放弃表或临时表的所有更改。如果 *lAllRows* 是“真” (.T.)，并且启用表缓冲，则放弃表或临时表中对所有记录所做的修改；如果 *lAllRows* 是“假” (.F.)，并且启用表缓冲，则只放弃对表或临时表中当前记录所做的修改。

如果启用行缓冲，则忽略 *lAllRows* 的值，并且放弃对表或临时表中当前记录所做的修改。

lAllRows 的默认值是“假” (.F.)。

cTableAlias

指定要放弃修改的表或临时表的别名。

nWorkArea

指定要放弃修改的表或临时表所在的工作区。

说明

TABLEREVERT () 函数返回放弃修改的记录数目。

注意 在网络上，当前存贮在磁盘上的数据可能和打开表或创建临时表时磁盘上的数据不同。这是因为在打开表或创建临时表之后，其他网络用户可能改变了数据。

对于没有启用行缓冲或表缓冲的表或临时表，TABLEREVERT () 函数不能放弃对它们的修改。如果不启用行缓冲或表缓冲而调用 TABLEREVERT () 函数，Visual FoxPro 将产生错误信息。

如果不用可选参数 *cTableAlias* 或 *nWorkArea* 调用 TABLEREVERT () 函数，则只放弃对当前工作区中打开表或临时表所做的修改。

示例

下面的示例演示了如何用 TABLEREVERT () 放弃对缓冲表做的更改。作为对表缓冲的要求，MULTILOCKS 应设置为 ON。打开数据库 testdata 中的 customer 表，然后用 CURSORSETPROP () 将缓冲模式设置成最佳表缓冲 (5) 。

显示 cust_id 字段的值，然后用 REPLACE 修改它。显示 cust_id 字段的新值。用 TABLEREVERT () 将表返回到最初状态 (TABLEREVERT () 代替了提交更改) 。

```
CLOSE DATABASES
SET MULTILOCKS ON && 表缓冲必须为 ON
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
= CURSORSETPROP ( 'Buffering', 5, 'customer' ) && 激活表缓冲
```

```
CLEAR
? 'Original cust_id value: '
?? cust_id && 显示 current cust_id 值
REPLACE cust_id WITH '***' && 改变字段内容
? 'New cust_id value: '
?? cust_id && 显示新 cust_id 值
= TABLEREVERT (.T.) && 放弃对表的修改
? 'Reverted cust_id value: '
?? cust_id && 显示恢复后的 cust_id 值
```

请参阅

[CURSORSETPROP \(\)](#) , [CURVAL \(\)](#) , [OLDVAL \(\)](#) , [TABLEUPDATE \(\)](#)

TABLEUPDATE () 函数

执行对缓冲行、缓冲表或临时表的修改。

语法

```
TABLEUPDATE ( [nRows [, lForce]] [, cTableAlias | nWorkArea] [, cErrorArray] )
```

返回值类型

逻辑型

参数描述

nRows

确定实施对表或临时表的哪些更改。如果 `nRows` 是 0 (或 `.F.`)，并且启用行或表缓冲，那么只实施对表或临时表中当前记录的更改。

如果 `nRows` 是 1，并且启用表缓冲，那么实施对表或临时表中所有记录所做的更改。如果 `nRows` 是 1 (或 `.T.`)，并且启用行缓冲，那么只实施对表或临时表中当前记录所做的更改。

如果 `nRows` 是 2，则对表或临时表实施的更改方式与 `nRows` 是 1 的情况相同。但是，当不能实施一处更改时，不产生一个错误，并且 Visual FoxPro 继续处理表或临时表中的其他记录。如果包含了 `cErrorArray`，则当产生一个错误时，创建一个包含错误信息的数组。

`nRows` 的默认值是 0。

lForce

决定是否覆盖网络上的其他用户对表或临时表的修改。如果 `lForce` 是“真” (`.T.`)，那么网络上其他用户表或临时表的任何更改都被覆盖。

如果 `lForce` 是“假” (`.F.`)，Visual FoxPro 从表或临时表的第一个记录开始，对记录逐条实施更改，直到表或临时表的结束。如果遇到某个记录被网络上别的用户修改过，Visual FoxPro 将产生错误。

当 Visual FoxPro 产生该错误时，可以用 `ON ERROR` 例程来处理。`ON ERROR` 例程调用 `TABLEUPDATE()` 函数，并把 `lForce` 设置成“真”

(`.T.`)，从而实施对记录的更改。或者，如果正处理一个事务，那么 `ON`

ERROR 例程可以处理发生的错误，并且调用 ROLLBACK 命令，把表或临时表恢复成原来的状态。

lForce 的默认值是“假”（.F.）。

cTableAlias

指定要实施更改的表或临时表的别名。如果包含了表或临时表的别名，也必须包含 *lForce* 参数。

nWorkArea

指定要实施更改的表或临时表所在的工作区。

cErrorArray

指定当 *nRows* 是 2，并且不能实施对一个记录的更改时，所创建的数组。该数组只包含一行，其中包含不能实施更改的记录的编号。如果包含了一个数组名，也必须包含表或临时表的别名 *cTableAlias* 或工作区编号 *nWorkArea*。

注意 如果在更新记录时发生了一个非简单的提交错误时，*cErrorArray* 的第一个元素将包含 -1，还可以运用 *AERROR*（）函数判断更新不被认可的原因。

说明

如果实施了对所有记录所做的更改，则 *TABLEUPDATE*（）函数返回“真”（.T.）；否则，*TABLEUPDATE*（）函数返回“假”（.F.），记录指针指向不能实施更改的记录。可以发出 *AERROR*（）函数确定对此记录所做更改不能实施的原因。

对于没有启用行缓冲的表或没有启用表缓冲的临时表，*TABLEUPDATE*（）函数不能实施对它们的更改。如果在没有启用行或表缓冲的情况下，调用 *TABLEUPDATE*（）

函数，Visual FoxPro 将产生错误信息。然而，TABLEUPDATE () 函数仍然可以对具有有效性规则的表或临时表实施更改。使用函数 CURSORSETPROP () 可以启用或废止行或表缓冲。

如果不带可选参数 *cTableAlias* 或 *nWorkArea* 调用 TABLEUPDATE () 函数，那么只对当前所选工作区中打开的表或临时表实施更改。

如果使用了表缓冲，并且更新了多个记录，则 TABLEUPDATE () 将记录指针移动到更新的最后一个记录。

注意 对于不使用关键字段的本地表或视图，调用 TABLEUPDATE () 会产生一个长 WHERE 子句，以查找更新的行。WHERE 子句中支持的字段默认数是 40。如果得到错误 1812 -SQL: 语句太长，可以为更新使用关键字段，或者用 SYS (3055) 来增加 WHERE 子句的复杂程度。如果使用 SYS (3055)，其值增加到表中字段数量的 8 倍。

= SYS (3055, 8 * MIN (40, FCOUNT ()))

示例

下面的示例演示了如何用 TABLEUPDATE () 提交对缓冲表所做的更改。创建表 employees，然后用 INSERT-SQL 在 cLastName 字段中插入值 “Smith”。

作为对表缓冲的要求，MULTILOCKS 应设置为 ON。用 CURSORSETPROP () 将缓冲模式设置成最佳表缓冲 (5)。

显示 cLastName 字段 (Smith) 的最初值，然后用 REPLACE 修改它。显示 cLastName 字段的新值 (Jones)。用 TABLEUPDATE () 提交更改到表中 (用 TABLEREVERT () 代替了放弃更改)。显示 cLastName 字段更新后的值 (Jones)。

```
CLOSE DATABASES
CREATE TABLE employee ( cLastName C ( 10) )
SET MULTILOCKS ON && 表缓冲必须为 ON
= CURSORSETPROP ( 'Buffering', 5, 'employee' ) && 激活表缓冲
INSERT INTO employee ( cLastName ) VALUES ( 'Smith' )
```

```
CLEAR
? 'Original cLastName value: '
?? cLastName && 显示 cLastName 当前值 ( Smith )
```

```
REPLACE cLastName WITH 'Jones'
? 'New cLastName value: '
?? cLastName && 显示 cLastName 新值 ( Jones )
```

```
= TABLEUPDATE ( .T. ) && 接受更新
? 'Updated cLastName value: '
?? cLastName && 显示 cLastName 当前值 ( Jones )
```

请参阅

[CURSORSETPROP \(\)](#) , [CURVAL \(\)](#) , [OLDVAL \(\)](#) , [TABLEREVERT \(\)](#)

Tabs 属性

指定页框控件中是否有选项卡。设计和运行时可用。

语法

PageFrame.Tabs[= *lExpr*]

参数描述

lExpr

Tabs 属性的设置有：

设置

说明

“真”

（默认值）页框有选项卡

（.T.）

“假”

页框没有选项卡

（.F.）

说明

通过设置页面的 Caption 属性，指定选项卡文本内容和宽度。

应用于

页框

请参阅

[Caption 属性](#)

_TABS 系统变量

包含制表符设置。包含此变量是为了提供向后兼容性，可使用报表设计器代替。

TabStop 属性

指定用户是否可以使用 TAB 键把焦点移动到对象上。设计和运行时可用。

语法

[object.]control.TabStop[= lExpr]

参数描述

lExpr

指定一个对象是否包含在 Tab 键次序中，TabStop 属性的设置有：

设置	说明
“真” (.T.)	(默认值) 控件或表单包含在 Tab 键次序中，Tab 键次序由 TabIndex 属性决定。

“假”
(.F.)

在 Tab 键次序中忽略该对象。对于一个控件来说，当用户按了 TAB 键，焦点将跳过属性设置为“假”(.F.) 的控件；对于表单，当用户在表单的最后一个控件处按下 TAB 键，而 Tab 键次序中下一个表单属性设置为“假”(.F.) 时，焦点将移到这个表单的第一个控件处，而不是移到 Tab 键次序中的下一个表单上。

说明

如果一个控件或表单的 TabStop 属性设置为“假”(.F.)，那么当按下 TAB 键在控件或表单中移动焦点时，将跳过这个控件或表单。但是当使用鼠标时，该控件或表单仍然可以接受焦点。

注意 当用于包含在列对象中的控件时，TabStop 属性只读。

应用于

复选框，组合框，命令按钮，容器对象，控件对象，编辑框，表单，表格，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，页面，页框，_SCREEN，微调，文本框

请参阅

[TabIndex 控件](#)

TabStretch 属性

指定当选项卡在页框控件中容纳不下时，页框的动作。设计和运行时可用。

语法

PageFrame.TabStretch[= *nSetting*]

参数描述

nSetting

TabStretch 属性的设置有：

设置

说明

0	堆积。创建第二行选项卡。
1	(默认值) 剪裁。按需要裁剪选项卡。

应用于

页框

请参阅

[Tabs 属性](#)

TabStyle 属性

指定页框中的页面选项卡是否调整。设计和运行时可用。

语法

```
PageFrame.TabStyle[ = nStyle ]
```

参数描述

nStyle

取下列一个值：

nStyle

说明

-
- | | |
|---|---|
| 0 | （默认值）调整。调整每个页面选项卡的宽度，以容纳标题。如果必要，增加每个页面选项卡的宽度，这样，页面选项卡会横跨页框的整个宽度。
如果 TabStretch 属性设置为 1 - 单行，会忽略这个设置。 |
| 1 | 不调整。不调整每个页面选项卡的宽度，以容纳标题。
如果 TabStretch 属性设置为 1 - 单行，会忽略这个设置。 |

应用于

页框

请参阅

[TabStretch 属性](#)

Tag 属性

存贮用户程序所需的任何其他数据。设计和运行时可用。

语法

Object.Tag[= *Expr*]

参数描述

Expr

指定任何字符串。

说明

默认情况下，Tag 属性设置为一个空字符串（“”）。

使用这个属性，可以在不产生任何影响的情况下，把任何字符串赋给一个对象。与其他属性不同的是，Visual FoxPro 并不使用 Tag 属性的值。

应用于

ActiveDoc 对象，复选框，组合框，命令按钮，命令组，编辑框，表单，表格，图像，标签，线条，列表框，OLE 绑定型控件，OLE 容器控件，选项按钮，项目挂接对象，选项组，_SCREEN，形状，微调，文本框，计时器

请参阅

[Comment 属性](#)

TAG () 函数

返回打开的 .CDX 多项复合索引文件的标识名，或者返回打开的 .IDX 单项索引文件的文件名。

语法

TAG ([*CDXFileName*,] *nTagNumber* [, *nWorkArea* | *cTableAlias*])

返回值类型

字符型

参数描述

CDXFileName

指定多项复合索引文件 .CDX 的文件名，TAG () 函数将返回该文件的一个标识名。

nTagNumber

nTagNumber 指定一个数值，此数值对应于复合索引文件中标识的创建顺序。例如，如果 nTagNumber 是 1，TAG () 函数将返回复合索引文件中创建的第一个标识名；如果 nTagNumber 是 2，TAG () 函数将返回所创建的

第二个标识名，依此类推。当 `nTagNumber` 超过标识总数时，`TAG()` 函数将返回一个空字符串。如果省略了 `CDXFileName` 参数，`TAG()` 函数将按以下特定顺序返回复合索引文件的标识名和单项索引文件的文件名：

1. 依据它们在 `SET INDEX` 或 `USE` 的 `INDEX` 子句中出现的次序，返回单项索引文件的文件名。
2. 返回结构复合索引文件（如果该表存在这样的文件）标识名。
3. 返回其他打开的复合索引文件中的标识名。同样是按标识在复合索引文件中的创建次序和复合索引文件在 `SET INDEX` 或 `USE` 的 `INDEX` 子句中的指定顺序返回。

`nWorkArea | cTableAlias`

从其他工作区中打开的文件中返回标识名或索引文件名。`nWorkArea` 指定工作区号，`cTableAlias` 指定表别名。默认情况下，返回当前工作区中文件的标识名和索引文件名。

说明

标识是多项复合索引 `.CDX` 文件的索引项。

注意 可以用 `INDEX` 语句创建多项复合索引 `.CDX` 文件。用 `SET INDEX` 或 `USE` 语句中的 `INDEX` 子句，可以为一个表打开多项复合索引文件（`.CDX`）和单项索引文件（`.IDX`）。结构复合索引文件随表的打开而自动打开。

示例

下面的示例打开 `testdata` 数据库中的 `customer` 表。使用 `FOR ... ENDFOR` 语句创建一个循环，在循环内使用 `CDX()` 显示每个结构索引的名称。

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

```
USE Customer && 打开“customer”表
CLEAR
FOR nCount = 1 TO 254
    IF !EMPTY (TAG (nCount) ) && 检查索引中的标识。
        ? CDX (nCount) && 显示 .CDX 名称
    ELSE
        EXIT && 当找不到标识时，退出循环
    ENDIF
ENDFOR
```

请参阅

[CDX \(\)](#) , [INDEX](#) , [MDX \(\)](#) , [NDX \(\)](#) , [SET INDEX](#) , [SYS \(14 \)](#) , [USE](#)

TAGCOUNT () 函数

返回复合索引文件 (.CDX) 标识以及打开的单项索引文件 (.IDX) 的数目。包含此函数是为了提供与 dBASE 的兼容性。

语法

```
TAGCOUNT ( [CDXFileName [, nExpression | cExpression]])
```

返回值类型

数值型

参数描述

CDXFileName

指定复合索引文件的文件名，TAGCOUNT () 函数返回其标识数目。如果省略了 *CDXFileName* 参数，TAGCOUNT () 函数将返回所有复合索引文件 (.CDX) 的标识数目，以及当前选定工作区中打开的单项索引文件 (.IDX) 总数。

nExpression

指定表的工作区，TAGCOUNT () 为此表返回复合索引文件 .cdx 的标识数量。

cExpression

指定表的别名，TAGCOUNT () 为此表返回复合索引文件 .cdx 的标识数量。

说明

包括与 dBASE 的兼容性。

请参阅

[INDEX, TAGNO \(\)](#)

TAGNO () 函数

返回复合索引文件 .cdx 标识和打开的单项索引文件 .idx 的索引位置。

语法

TAGNO ([IndexName [, CDXFileName [, nExpression | cExpression]])

返回值类型

数值型

参数描述

IndexName

指定复合索引文件 (.CDX) 标识名称或打开的单项索引文件 (.IDX) 的文件名, TAGNO () 函数返回该索引文件的索引位置。

CDXFileName

指定复合索引文件 (.CDX) 的文件名, 该文件包含 *IndexName* 变量中指定的标识名。

nExpression

指定表的工作区, TAGNO () 为此表返回复合索引文件 .cdx 标识和单项索引文件 .idx 的索引位置。

cExpression

指定表的别名, TAGNO () 为此表返回复合索引文件 .cdx 标识和单项索引文件 .idx 的索引位置。

说明

如果没有设置主控索引, 并且省略了可选参数, TAGNO () 函数将返回 0。

包括与 dBASE 的兼容性。

请参阅

INDEX, TAGCOUNT ()

_TALLY 系统变量

包含最近执行的表命令处理过的记录数目。

语法

`_TALLY = nRecords`

参数描述

nRecords

包含一个数值，该数值代表最近执行的表命令所处理的记录数目。

说明

某些表处理命令，在执行时将返回有关的状态信息（“对话”）。当这样的命令结束时，将显示所处理的记录数（如果 SET TALK 为 ON），并将这个数字存入 _TALLY 系统变量中。

下列命令返回状态信息：

APPEND	AVERAGE	CALCULATE
FROM		
COPY To	COUNT	DELETE
INDEX	PACK	REINDEX

REPLACE
SUM

SELECT - SQL
TOTAL

SORT
UPDATE

当启动 Visual FoxPro 时，_TALLY 设置为 0。执行以上的一个命令之后，_TALLY 值就变为该命令所处理的记录数目。

示例

下面的示例使用 SELECT 语句返回美国顾客数目。结果自动存入 _TALLY 系统变量中，然后显示 _TALLY 的值。

```
SELECT * FROM customer ;  
  WHERE country = 'USA' ;  
  INTO CURSOR temp  
? _TALLY
```

请参阅

SET ODOMETER, SET TALK

TAN () 函数

此函数返回角度的正切值。

语法

TAN (*nExpression*)

返回值类型

数值型

参数描述

nExpression

以弧度为单位指定一个角度，TAN () 函数返回该角度的正切值。若要把角度由度转换成弧度，可使用 DTOR () 函数。可以用 SET DECIMALS 语句指定 TAN () 函数返回数值的小数位。

示例

```
CLEAR
? TAN ( 0 )           && 显示 0.00
? TAN ( PI ( ) / 4 )  && 显示 1.00
? TAN ( PI ( ) * 3 / 4 ) && 显示 -1.00
```

请参阅

[COS \(\)](#) , [DTOR \(\)](#) , [RTOD \(\)](#) , [SET DECIMALS](#) , [SIN \(\)](#)

TARGET () 函数

返回一个表的别名，该表是 SET RELATION 命令的 INTO 子句所指定关系的目标。

语法

TARGET (*nRelationshipNumber* [, *nWorkArea* | *cTableAlias*])

返回值类型

字符型

参数描述

nRelationshipNumber

指定一个关系的编号，范围是从 1 到指定工作区与外部关系的总数目。当 *nRelationshipNumber* 不大于关系总数目时，返回目标数据库的别名。当 *nRelationshipNumber* 大于关系总数目时，返回一个空串。

nWorkArea | *cTableAlias*

返回另一个工作区中目标表的别名。 *nWorkArea* 指定工作区号， *cTableAlias* 指定表的别名。

示例

在下面的示例中，在 *order_id* 字段上设置从表 *orders* 到表 *customer* 的关系，然后用

TARGET () 函数返回目标表的别名。

```
CLOSE DATABASES
OPEN DATABASE (HOME (2) + 'data\testdata')
USE orders IN 0 ORDER order_id
USE customer IN 0 ORDER cust_id
SELECT orders
SET RELATION TO cust_id INTO customer
? TARGET (1)    && 显示 "customer"
? TARGET (2)    && 显示空串
```

请参阅

RELATION () , SET RELATION

TerminateRead 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。

Text 属性

包含在控件的文本框部分中输入的未设置格式的文本。设计时不可用；运行时只读。

语法

Object.Text

说明

与 Value 控件不一样，Text 控件包含的值是未设置格式的，并且与用户在控件中输入的文本一致。

控件的 Text 控件中的未设置格式的文本可能与控件的 Value 控件不一样，有以下原因：

- Value 控件可能不是字符型；可能包含日期或数值型

- 对编辑框来说，Value 控件可能会为在编辑框中输入的文本添加行，并提供与前期版本的兼容性。因此，在选择带有 SelStart 和 SelLength 属性的文本时，使用 Text 控件是更好的方法。
- 如果控件的 Format 控件用 R 设置，则从 Value 控件中删除输入的掩码字符。

如果组合框控件的 Style 控件设置为 2（下拉列表），因为组合框没有文本框，所以 Text 控件包含空串。

应用于

组合框，编辑框，文本框，微调

请参阅

Format 控件, SelStart 属性, SelLength 属性, Style 属性, Value 属性

TEXT ... ENDTXT 命令

输出文本行、表达式和函数的结果与变量的内容。

语法

TEXT

TextLines

ENDTXT

参数描述

TextLines

指定传送到当前输出设备的文本。TextLines 可由文本、变量、数组元素、表达式、函数或它们的任意组合构成。

只有在 SET TEXTMERGE 为 ON，并且被 SET TEXTMERGE DELIMITERS 指定的分隔符包围时，TextLines 指定的表达式、函数、变量和数组元素才能进行计算。如果 SET TEXTMERGE 为 OFF，表达式、函数、变量和数组元素输出为与分隔符一起的原义字符。

例如，SET TEXTMERGE 为 ON 并且文本行包含 <<DATE () >> 时，输出今天的日期。如果 SET TEXTMERGE 为 OFF，<<DATE () >> 会原义输出。

如果在 TEXT 和 ENDTEXT 中或者在 \ 或 \\ 后加上注释，则注释也被输出。

说明

结构化的程序命令将 TEXT 和 ENDTEXT 之间的文本行传送到 VisualFoxPro 主窗口、用户自定义窗口、打印机、文本文件或低级文件。

TEXT 传送文本行到当前输出设备中。这个过程将一直继续下去，直到 ENDTEXT 语句或程序结束。

默认情况下,TEXT...ENDTEXT 的输出将传送到 VisualFoxPro 主窗口或活动窗口中。SET CONSOLE OFF 命令可以禁止输出到 VisualFoxPro 主窗口或活动窗口。用 SET PRINTER 可将输出传送到打印机或文本文件中。

TEXT...ENDTEXT 的输出也可以传送到用 FCREATE ()、FOPEN () 创建或打开的低级文件中。如果将 FCREATE () 或 FOPEN () 返回的文件句柄保存到 _TEXT 系统变量中，输出会直接传到相应的低级文件。

示例

下面的示例演示了如何使用 SET TEXTMERGE、SET TEXTMERGE DELIMITERS、TEXT...ENDTEXT， 和 the _TEXT 系统变量。

创建低级文件 Names.txt，并在系统变量 _TEXT 中保存它的文件句柄。如果不能创建 Names.txt，则退出程序。打开 customer 表，前 10 个合同的名称输出到 Names.txt 中。输出文本和函数的结果到文本文件中。

用 MODIFY FILE 打开包含此名称的文本文件。

```
CLEAR
CLOSE DATABASES
SET TALK OFF
SET TEXTMERGE ON    && 激活函数或表达式的嵌入
STORE FCREATE ('names.txt') TO _TEXT    && 创建一个低级文件
IF _TEXT = -1 && 不能创建，退出程序
    WAIT WINDOW 'Cannot create an output file; press a key to exit'
    CANCEL
ENDIF

CLOSE DATABASES
OPEN DATABASE (HOME (2) + 'Data\testdata')
USE customer && 打开 Customer 表

TEXT
    CONTACT NAMES
    <<DATE ( ) >>    <<TIME ( ) >>
ENDTEXT
WAIT WINDOW 'Press a key to generate the first ten names'
SCAN NEXT 10
TEXT
    <<contact>>
```

```
ENDTEXT
ENDSCAN
CLOSE ALL && 关闭表和文本文件
MODIFY FILE names.txt
ERASE names.txt
```

请参阅

[FOPEN \(\)](#), [_PRETEXT](#), [SET TEXTMERGE](#), [SET TEXTMERGE DELIMITERS](#), [_TEXT](#)

_TEXT 系统变量

把 “\|\\” 和 TEXT ... ENDTEXT 文本合并命令的结果输出到低级文件中。

语法

```
_TEXT = nFileHandle
```

参数描述

nFileHandle

指定一个数值，该数值确定输出的低级文件。

说明

\|\\ 和 TEXT ... ENDTEXT 能很容易地把表的内容、变量以及函数和表达式的结果合并到文本中。由这些文本合并命令所产生的结果可以输出到屏幕或窗口或低级文件

中，_TEXT 系统变量可以把文本合并命令的结果输出到低级文件中。要把 “\| \\\” 和 TEXT ... ENDTEXT 文本合并命令的结果输出到低级文件中，需要在 SET TEXTMERGE 命令中加入 TO *fileName* 子句，*fileName* 的文件句柄保存在 _TEXT 变量中。如果在 _TEXT 中已存有另一个低级文件的句柄，则关闭该低级文件。用 FCREATE () 函数创建并打开低级文件，用 FOPEN () 函数打开已存在的低级文件。当文件成功地创建或打开时，这两个函数返回一个正的文件句柄。把该文件句柄保存在 _TEXT 中，将把 “\| \\\” 和 TEXT ... ENDTEXT 文本合并命令的任意结果连续输出到该文件中。使用 STORE 或 = 可以把文件句柄存入 _TEXT 中。打开的低级文件必须具有写特权，以便接收文本合并命令的输出。可以使用 FCLOSE () 或 CLOSE ALL 关闭低级文件，也可以用 SET TEXTMERGE TO 命令而不加文件名来关闭低级文件，该文件的句柄保存在 _TEXT 中。_TEXT 的初始默认值是 -1。如果您正将文本合并输出到文件中，该文件的句柄存在 _TEXT 中，那么可以把 -1 存入 _TEXT 中，这样可以关闭对该文件的输出而不关闭该文件。通过向 _TEXT 中保存入不同的文件句柄和 -1，可以把文本合并输出到不同的文件中。

示例

下面的示例演示了如何把文本合并输出到不同的文件中。

```
SET TALK OFF
SET TEXTMERGE ON NOSHOW && 启用文本合并，不输出到屏幕上。
SET TEXTMERGE DELIMITERS TO && 默认的文本合并分隔符 <<, >>
SET TEXTMERGE TO date.txt && 创建并输出到 date.txt 文件中
STORE _TEXT TO gcDateHandle && 保存 date.txt 的文件句柄
STORE -1 TO _TEXT && 关闭对 date.txt 文件的输出，但保持 date.txt 文件打开
SET TEXTMERGE TO time.txt && 创建并输出到 time.txt 文件中
```

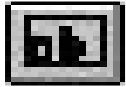


```
STORE _TEXT TO gcTimeHandle && 存贮 time.txt 的文件句柄
*** 向 time.txt 传送下列文本 ***
\The time is:
STORE gcDateHandle TO _TEXT && 现在把文本输出到 date.txt 文件
*** 向 date.txt 传送下列文本 ***
\Today's date is:
STORE gcTimeHandle TO _TEXT && 现在把文本输出到 time.txt 文件
*** 把时间输出在同一行上 ***
\\ <<TIME ( ) >>
STORE gcDateHandle TO _TEXT&& 现在把文本输出到 time.txt 文件
*** 把日期输出在同一行上 ***
\\ <<DATE ( ) >>
CLOSE ALL && 关闭所有文件
TYPE date.txt && 查看 date.txt 文件的内容 ...
WAIT WINDOW && 暂停
TYPE time.txt && ...以及 time.txt 文件的内容
ERASE date.txt
ERASE time.txt
```

请参阅

[\\, FOPEN \(\) , FCLOSE \(\) , FCREATE \(\) , _PRETEXT, SET TEXTMERGE, SETTEXTMERGE DELIMITERS](#)

TextBox 控件



创建一个文本框。

语法

TextBox

说明

创建一个文本框，从中可以编辑变量、数组元素或字段的内容。所有标准的 Visual FoxPro 编辑功能，例如剪切、复制和粘贴在文本框中都可用。

使用输入掩码和 Format 属性，可以指定在文本框中如何输入和显示数值。

有关创建文本框的详细内容，请参阅帮助中的“表单设计器”，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

BackColor

BackStyle

BaseClass

BorderColor

BorderStyle

Century

Class

续表

ClassLibrary	ColorScheme	ColorSource
Comment	ControlSource	DateFormat
DateMark	DisabledBackColor	DisabledForeColor
DragIcon	DragMode	Enabled
FontBold	FontCondense	FontExtend
FontItalic	FontName	FontOutline
FontShadow	FontSize	FontStrikeThru
FontUnderline	ForeColor	Format
Height	HelpContextID	HideSelection
Hours	IMEMode	InputMask
IntegralHeight	Left	Margin
MaxLength	MemoWindow	MouseIcon
MousePointer	Name	NullDisplay
OLEDragMode	OLEDragPicture	OLEDropEffects
OLEDropHasData	OLEDropMode	OLEDropTextInsertion
OpenWindow	Parent	ParentClass

续表

PasswordChar	ReadOnly	RightToLeft
Seconds	SelectedBackColor	SelectedForeColor
SelectOnEntry	SelLength	SelStart
SelText	SpecialEffect	StatusBarText
StrictDateEntry	Style	TabIndex
TabStop	Tag	TerminateRead
Text	ToolTipText	Top
Value	Visible	WhatsThisHelpID
Width		
事件		
Click	DbClick	Destroy
DragDrop	DragOver	Error
ErrorMessage	GotFocus	Init
InteractiveChange	KeyPress	LostFocus
Message	MiddleClick	MouseDown
MouseMove	MouseUp	MouseWheel

OLECompleteDrag

OLEDragDrop

OLEDragOver

OLEGiveFeedBack

OLESetData

OLEStartDrag

ProgrammaticChange

RangeHigh

RangeLow

RightClick

UIEnable

Valid

When

方法

AddProperty

CloneObject

Drag

Move

OLEDrag

ReadExpression

ReadMethod

Refresh

ResetToDefault

SaveAsClass

SetFocus

ShowWhatsThis

WriteExpression

WriteMethod

ZOrder

请参阅

CREATE CLASS, CREATE FORM, DEFINE CLASS

TextHeight 方法

返回以当前字体显示的文本字符串高度。

语法

```
[nHeight = ] [FormSet.]object.TextHeight ( cText )
```

参数描述

nHeight

返回一个数值，该数值指定文本字符串的高度。

cText

指定要确定高度的字符串。高度的单位由表单的 ScaleMode 属性指定。

说明

TextHeight 方法确定显示 *cText* 所需要的垂直空间。所返回的高度包括字符串上下的前导空间，因此可以使用返回的高度来计算和放置表单上的多个文本行。如果 *cText* 中包含嵌入的回车符，那么 TextHeight 方法将返回文本行的累计高度，包括每行上下的前导空间。

应用于

表单，_SCREEN

请参阅

[FontSize 属性](#)，[ScaleMode 属性](#)，[TextWidth 方法](#)

TextWidth 方法

返回以当前字体显示的文本字符串宽度。

语法

```
[nWidth = ] [FormSet.]object.TextWidth ( cText)
```

参数描述

nWidth

返回一个数值，该数值指定文本字符串的宽度。

cText

指定要确定宽度的字符串。宽度的度量单位由表单的 ScaleMode 属性指定。

说明

TextWidth 方法用来确定显示 *cText* 所需要的水平空间。

应用于

表单，_SCREEN

请参阅

[FontSize 属性](#)，[ScaleMode 属性](#)，[TextHeight 方法](#)

THIS 对象

在创建对象之前，提供对它的引用。

语法

`THIS.Property Name | Object Name`

参数描述

`PropertyName`

指定类的属性。

`ObjectName`

指定类中的对象。

说明

THIS 允许在一个类定义中引用属性或对象。类定义块中的方法可以使用 THIS 来指定类创建时存在的属性或对象。

因为对象的多个实例共享同样的方法代码，因此 THIS 总是指代码正在执行的实例。如果存在一个对象的多个实例，并且调用了对象的方法，THIS 会引用正确的对象。

示例

下面的示例在 Form 类的基础上创建了子类 MyForm。创建了 ChangeBackColor 方法。

ChangeBackColor 通过 THIS 来引用 MyForm。

```
DEFINE CLASS MyForm AS FORM
```



```
CAPTION = "This Form"  
HEIGHT = 15  
WIDTH = 20  
  
PROCEDURE ChangeBackColor  
PARAMETER NewColor  
    THIS.BACKCOLOR = NewColor  
ENDPROC  
ENDDEFINE
```

请参阅

[THISFORM](#), [THISFORMSET](#)

THISFORM 对象

在一个表单创建之前，提供对它的引用。

语法

```
THISFORM. .PropertyName | ObjectName
```

参数描述

PropertyName

指定表单的属性。

ObjectName

指定表单中的对象。

说明

THISFORM 提供了一种指向正在写入事件处理程序的表单的简便的方法。例如，对命令按钮的 Click 事件程序设置了按钮对当前时间的说明：

```
thisform.command1.caption = time ( )
```

使用 THISFORM 代替按姓名明确地指向表单使程序代码更加简便。

当创建类定义时，THISFORM 也可以指向带有方法的当前表单。THISFORM 允许引用表单上的对象或表单的属性，而不需要使用多个 Parent 属性。

请参阅

[THIS](#), [THISFORMSET](#)

THISFORMSET Object Reference

在一个表单集创建之前，提供对它的引用。

语法

`THISFORMSET.PropertyName | ObjectName`

参数描述

PropertyName

指定表单集的属性。

ObjectName

指定表单集中的对象。

说明

THISFORMSET 提供了在方法中对对象所在表单集或表单集属性的引用。

THISFORMSET 允许引用表单集上的对象或表单集的属性，而不需要使用多个 Parent 属性。

请参阅

[THIS, THISFORM](#)

_THROTTLE 系统变量

当跟踪窗口打开时，指定程序的执行速度。

语法

`_THROTTLE = nSeconds`

参数描述

nSeconds

指定执行每个程序行所间隔的秒数。初始默认值是 0，程序行的执行没有停

顿。 *nSeconds* 可以从 0 到 5.5 秒。例如，如果 *nSeconds* 是 0.5 秒，那么每执行一行程序后都要延迟 1/2 秒。

说明

跟踪窗口是 Visual FoxPro 中的调试工具，它在执行时显示程序的源代码。跟踪窗口突出显示每一正在执行的程序行。THROTTLE 的数值决定程序行之间的执行间隔。

注意

要想延迟程序的执行，跟踪窗口“程序”菜单的“断续跟踪”选项必须为打开。

请参阅

SET DEBUG, SET STEP

TIME () 函数

以 24 小时制、8 位字符串（时：分：秒）格式返回当前系统时间。

语法

TIME ([*nExpression*])

返回值类型

字符型

参数描述

nExpression

如果含有 *nExpression* 变量，可以返回包括百分之一秒的时间。数值表达式 *nExpression* 可以是任何值，然而，实际的最大精度值是 1/18 秒。使用 SECONDS () 函数可获得更高的精度。

请参阅

CTOT () , DATE () , DATETIME () , DTOT () , HOUR () , SEC
() , SECONDS () , SETSECONDS , SYS (2)

Timer 控件



创建以一定时间间隔执行代码的计时器。

语法

Timer

说明

计时器控件对用户是不可见的，它对于后台处理很有用。计时器的典型应用是检查系统时钟，决定是否到了某个程序或应用程序运行的时间。

有关创建计时器的详细内容，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	BaseClass	Class
ClassLibrary	Comment	Enabled
Height	Interval	Left
Name	Parent	ParentClass
Tag	Top	Width

事件

Destroy	Error	Init
Timer		

方法

AddProperty	Cloneobject	ReadExpression
ReadMethod	Reset	ResetToDefault
SaveAsClass	WriteExpression	WriteMethod

请参阅

CREATE CLASS, CREATE FORM, DEFINE CLASS, 表单设计器

Timer 事件

当经过 Interval 属性中指定的毫秒数时，此事件发生。

语法

```
PROCEDURE Timer.Timer  
[LPARAMETERS nIndex]
```

参数描述

nIndex

唯一标识控件数组中的一个控件。

应用于

计时器

请参阅

[Interval 控件](#), [Reset 方法](#)

TitleBar 属性

指定在一个表单的顶部是否显示工具栏。

语法

```
object.TitleBar[ = nExpression]
```

参数描述

nExpression

TitleBar 属性的设置有：

设置

说明

0

不显示工具栏。

1

(默认值) 显示工具栏。

说明

如果 TitleBar 属性设置为 0，则不显示工具栏，并且窗口弹出菜单，以及最大、最小化和关闭按钮都不可用。

应用于

表单

请参阅

[Closable 属性](#) , [MaxButton 属性](#) , [MinButton 属性](#) , [WhatsThisButton 属性](#)

ToolBar 对象

创建自定义工具栏。

语法

ToolBar

说明

使用工具栏对象可以为应用程序创建工具栏。

下表列举了自定义工具栏的属性：

- 工具栏总是位于顶部。
- 当被移动到任何 Visual FoxPro 主窗口边界时，工具栏可以自动停放。
- 当工具栏没有停放的时候，以一个带有标题的半高栏出现。
- 当改变工具栏的尺寸时，控件也相应做调整。
- 在工具栏任何没有控件的区域单击、拖动可以移动工具栏。
- 工具栏中的多数控件在被选中时，不能接受焦点。
- 工具栏中控件的访问键是无效的。

虽然工具栏中可以放入任何控件，但是象列表框这样的控件可能太大，以致不能合适地安放在已停放的工具栏中。可以编程方式在停放工具栏时将这些控件移去，并用同样的较小的控件或者不同的控件来代替。

有关创建工具栏的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十一章“设计菜单与工具栏”。

属性

Activecontrol	Application	BackColor
BaseClass	Caption	Class
ClassLibrary	Comment	ColorSource
ControlBox	ControlCount	Controls
DataSession	DataSessionID	Docked
DockPosition	Enabled	ForeColor
Height	HelpContextID	KeyPreview
Left	LockScreen	MouseIcon
MousePointer	Movable	Name
Objects	OLEDragMode	OLEDragPicture
OLEDropEffects	OLEDropHasData	OLEDropMode
Parent	ParentClass	ScaleMode
ShowTips	ShowWindow	Sizable
Tag	Top	Visible
WhatsThisHelpID	Width	

事件

Activate	AfterDock	BeforeDock
Click	Db1Click	Deactivate

续表

Destroy	DragDrop	DragOver
Error	Init	MiddleClick
MouseDown	MouseMove	MouseUp
MouseWheel	Moved	OLECompleteDrag
OLEDragDrop	OLEDragOver	OLEGiveFeedBack
OLESetData	OLEStartDrag	Paint
Resize	RightClick	UnDock

方法

Addobject	AddControl	Cloneobject
Dock	Hide	Move
Newobject	OLEDrag	ReadExpression
ReadMethod	Refresh	Release
Removeobject	ResetToDefault	SaveAsClass
SetAll	Show	ShowWhatsThis
WriteExpression	ZOrder	

示例

下面的示例演示了如何通过 Toolbar 类创建工具栏。在调整工具栏大小时，Visual FoxPro 会重新安排按钮。创建了 ToolBar 对象后，Visual FoxPro 自动地以添加到类定义的顺序，将控件从左到右进行放置，同时忽略控件的 Top 和 Left 属性。

```
PUBLIC tbrDesktop  
tbrDesktop = CREATEOBJ ('mytoolbar')
```

```
tbrDesktop.SHOW
DEFINE CLASS myToolBar AS Toolbar
  ADD object btnBold AS CommandButton
  ADD object sep1 AS Separator
  ADD object btnItalics AS CommandButton
  btnBold.HEIGHT = 20
  btnBold.WIDTH = 50
  btnBold.Caption = "Bold"
  btnItalics.HEIGHT = 20
  btnItalics.WIDTH = 50
  btnItalics.Caption = "Italic"
  btnItalics.FontBold = .F.
  LEFT = 1
  TOP = 1
  WIDTH = 25
  CAPTION = "Desktop Attributes"
  PROCEDURE Activate
    this.btnBold.FontBold = _SCREEN.FONTBOLD
    this.btnItalics.FontItalic = _SCREEN.FONTITALIC
  ENDPROC
  PROCEDURE btnBold.CLICK
    _SCREEN.FONTBOLD = !_SCREEN.FONTBOLD
    This.FontBold = _SCREEN.FONTBOLD
  ENDPROC
  PROCEDURE btnItalics.CLICK
    _SCREEN.FONTITALIC = !_SCREEN.FONTITALIC
    This.FontItalic = _SCREEN.FONTITALIC
  ENDPROC
ENDDEFINE
```

请参阅

CREATE CLASS, CREATE FORM, DEFINE CLASS, Separator 对象

ToolTipText 属性

为一个控件指定作为“工具提示”出现的文本。设计和运行时可用。

语法

```
control.ToolTipText = cText
```

参数描述

cText

指定“工具提示”所使用的文本。可设定的文本字符的最大值为 127。

说明

只有当包含控件的表单或工具栏的 ShowTips 属性设置为“真” (.T.) 时，才可以设置 ToolTipText 属性。

应用于

复选框，组合框，命令按钮，编辑框，表格，列表框，选项按钮，形状，微调，文本框

请参阅

ShowTips 属性

Top 属性

对于控件，指定相对其父对象最顶端的边缘所在位置；对于表单对象，确定表单顶端边缘与 Visual FoxPro 主窗口的距离。设计和运行时可用。

语法

object.Top[= *nValue*]

参数描述

nValue

对于控件，指定对象的上边缘和父对象的上边缘之间的距离。

对于表单对象，确定表单的上边缘和 Visual FoxPro 主窗口之间的距离。

说明

可以使用 Left, Top, Height 和 Width 属性改变对象的外部特征，例如移动对象和调整对象大小。

注意 当用于一个列对象中的控件时，Top 属性是只读的。

Top 属性的度量单位由控件的 ScaleMode 属性设置值决定。

示例

下面的示例演示了如何使用 Top 属性在表单中放置控件。用 Add 对象方法向表单中加入一个线条控件和三个命令按钮，

Top 属性指定表单中每个控件的垂直位置。

```
frmMyForm = CREATEobject ('Form')  && 创建一个表单  
frmMyForm.Closable = .F.  && 废止控件菜单框
```

```
frmMyForm.Addobject ('shpLine','Line')  && 向表单中加入 一个线条控件  
frmMyForm.Addobject ('cmdCmndBtn1','cmdMyCmndBtn1')  && “Up” 命令按钮  
frmMyForm.Addobject ('cmdCmndBtn2','cmdMyCmndBtn2')  && “Down” 命令按钮  
frmMyForm.Addobject ('cmdCmndBtn3','cmdMyCmndBtn3')  && “Quit” 命令按钮
```

```
frmMyForm.shpLine.Visible = .T.  && 使线条控件可见  
frmMyForm.shpLine.Top = 20  && 指定线条控件行  
frmMyForm.shpLine.Left = 125  && 指定线条控件列
```

```
frmMyForm.cmdCmndBtn1.Visible = .T.  && “Up” 命令按钮可见  
frmMyForm.cmdCmndBtn2.Visible = .T.  && “Down” 命令按钮可见  
frmMyForm.cmdCmndBtn3.Visible = .T.  && “Quit” 命令按钮可见
```

```
frmMyForm.SHOW  && 显示表单  
READ EVENTS  && 开始事件处理  
DEFINE CLASS cmdMyCmndBtn1 AS 命令 BUTTON  && 创建命令按钮  
    Caption = 'Slant \<Up'  && 命令按钮的标题  
    Left = 50  && 命令按钮的列  
  
    Top = 100  && 命令按钮的行  
    Height = 25  && 命令按钮的高度  
  
PROCEDURE Click  
    ThisForm.shpLine.Visible = .F.  && 隐藏线条控件
```

```
        ThisForm.shpLine.LineSlant = '/' && 向上倾斜
        ThisForm.shpLine.Visible = .T. && 显示线条控件
    ENDDDEFINE
```

```
DEFINE CLASS cmdMyCmndBtn2 AS CommandButton && 创建命令按钮
    Caption = 'Slant \<Down' && 命令按钮上的标题
    Left = 200 && 命令按钮的列
    Top = 100 && 命令按钮的行
    Height = 25 && 命令按钮的高度
```

```
    PROCEDURE Click
        ThisForm.shpLine.Visible = .F. && 隐藏线条控件
        ThisForm.shpLine.LineSlant = '\' && 向下倾斜
        ThisForm.shpLine.Visible = .T. && 显示线条控件
    ENDDDEFINE
```

```
DEFINE CLASS cmdMyCmndBtn3 AS 命令 Button && 创建命令按钮
    Caption = '\<Quit' && 命令按钮的标题
    Cancel = .T. && 默认的取消命令按钮 (ESC)
    Left = 125 && 命令按钮的列
    Top = 150 && 命令按钮的行
    Height = 25 && 命令按钮的高度
```

```
    PROCEDURE Click
        CLEAR EVENTS && 停止事件处理，关闭表单
    ENDDDEFINE
```

应用于

复选框，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，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，页框，_SCREEN，形状，微调，文本框，计时器，工具栏

请参阅

[Left 属性](#), [Height 属性](#), [Move 方法](#), [ScaleMode 属性](#), [Width 属性](#)

TopIndex 属性

指定出现在列表最顶端位置的列表项。设计时不可用，运行时可读写。

语法

```
control.TopIndex[ = nIndex]
```

参数描述

nIndex

指定出现在列表最上部的数据项的索引。

说明

数据项的索引反映了它在列表中的位置。该索引相对于列表中的其他数据项，并且可以更改。

应用于

组合框，列表框

请参阅

[AddItem 方法](#), [Clear 方法](#), [List 属性](#), [ListCount 属性](#), [MultiSelect 属性](#),

[NewItemID 属性](#), [RemoveItem 方法](#), [Selected 属性](#), [TopItemID 属性](#)

TopItemID 属性

指定出现在列表最上部的数据项标识。设计时不可用，运行时可读写。

语法

```
control.TopItemID[ = nItemID ]
```

参数描述

nItemID

指定出现在列表最上部的数据项标识。

说明

默认的 TopItemID 属性设置值为 1，即列表中的第一项。

应用于

组合框，列表框

请参阅

[AddItem 方法](#), [Clear 方法](#), [List 属性](#), [ListCount 属性](#), [MultiSelect 属性](#), [NewItemID 属性](#), [RemoveItem 方法](#), [Selected 属性](#), [TopIndex 属性](#)

TOTAL 命令

计算当前选定表中数值字段的总和。

语法

```
TOTAL TO TableName  
ON FieldName  
  [FIELDS FieldNameList]  
  [Scope]  
  [FOR lExpression1]  
  [WHILE lExpression2]  
  [NOOPTIMIZE]
```

参数描述

TableName

指定存放计算结果的表的名称。如果指定的表不存在，Visual FoxPro 将创建它；如果表存在，并且 SET SAFETY 为 ON，则 Visual FoxPro 将询问是否要改写这个已存在的表。如果 SET SAFETY 为 OFF，则不做任何提示直接改写输出表。

FieldName

指定总计时作为分组依据的字段。表必须以该字段排序，或者打开的索引或索引标识必须以该字段作为其关键字表达式。

FIELDS *FieldNameList*

指定要总计的字段。列表中的字段名用逗号分隔。如果省略了 FIELDS 子句，则默认合计所有的数值型字段。

Scope

指定要总计的记录范围。Scope 子句有：ALL、NEXT *nRecords*、Record *nReordNumber* 和 REST。

TOTAL 命令默认的范围是全部（ALL）记录。

FOR *lExpression1*

指定一个条件，只有满足该逻辑条件 *lExpression1* 的记录包含在总计中。

如果 *lExpression1* 是可优化表达式，那么 Rushmore 可以优化 TOTAL ... FOR 创建的查询。为了达到更好的性能，请在 FOR 子句中使用可优化表达式。

WHILE *lExpression2*

指定一个条件，只要逻辑表达式 *lExpression2* 计算为“真”（.T.）时，当前表中的记录就包含在总计中。

NOOPTIMIZE

关闭 TOTAL 的 Rushmore 优化。

说明

要使用此命令，当前工作区中的表必须经过排序或索引。对于具有相同字段值或索引关键字值的各组记录，将分别计算其总计值。总计结果放入另一个表的记录中，同时还将在此表中对这些字段值或索引关键字创建一条记录。

如果第二个表中数值字段的宽度不足以放置总计值，将会发生数值溢出错误。当发生数值溢出错误时，Visual FoxPro 保存总计值最主要的部分：

- 小数位被截断，即对总计值余下小数位进行圆整。

- 如果总计值仍然不能放下，例如包含 7 位以上的数字，这时将采用科学计数法表示。
- 最后，用星号代替字段的内容。

请参阅

AVERAGE, CALCULATE, INDEX, SORT, SUM

TRANSFORM () 函数

用 @ ... SAY 中的 PICTURE 或 FUNCTION 代码格式化字符表达式或数值表达式。

语法

TRANSFORM (*eExpression*, *cFormatCodes*)

返回值类型

字符型

参数描述

eExpression

指定要进行格式编排的数值或字符表达式。

cFormatCodes

指定一个或多个 PICTURE 或 FUNCTION 代码来决定如何对表达式进行格式编排。

格式代码

说明

@ C	CR 附加于正的货币型值或数值型值后面，表明该值是贷款。
@ D	日期和日期时间值转换为 SET DATE 当前设置的格式。
@ E	日期和日期时间值转换为 BRITISH 日期格式。
@ R	使用格式掩码进行转化。掩码字符不保存到转化后的值。只用于字符型或数值型数据。掩码字符包括：
9 or #	代表一个字符或数值。
!	将小写字母转化为大写字母。
@ T	从字符值中删除前导和后续空格。
@ X	DB 附加于负的货币型值或数值型值后面，表明该值是借款。
@ Z	如果为 0，则将货币型值或数值型值转化为空格。
@ (将负的货币型值或数值型值放在括号中。
@ ^	将货币型值或数值型值转化为科学计数格式。
@ 0	将货币型值或数值型值转化为等价的十六进制格式。货币型值或数值型值必须是正的，并且小于 4,294,967,296。
@ !	将整个字符串转化为大写。
@ \$	将当前的货币符号添加到货币型值或数值型值，当前的货币符号是由 SET CURRENCY 指定的。在默认情况下，该符号紧放在该值的前面或后面。但是，当前的货币符号和位置（由 SET CURRENCY 指定）、分隔字符（由 SET SEPARATOR 指定）和进制字符（由 SET POINT 指定）都有可能改变。
X	指定字符值的宽度。例如，如果 cFormatCodes 为 'XX'，则返回两个字符。
Y	将逻辑“真”（.T.）和“假”（.F.）分别转化为 Y 和 N。
!	将一个小写字母在字符串的相应位置转化为大写。
.	指定货币型值或数值型值中的小数点位置。

续表

， 分隔货币型值或数值型值中小数点左方的数字。

如果省略 cFormatCodes，则 Visual FoxPro 对 eExpression 进行默认的转换。下表列出了对 eExpression 的每种数据类型的转换：

数据类型	转换说明
字符型	不进行转换。
货币型	所进行的转换由“选项”对话框“地区”选项卡中的设置决定。
日期型	对日期值进行 DTOC () 转换。
日期时间型	对日期时间值进行 TTOC () 转换。
逻辑型	将逻辑“真” (.T.) 和“假” (.F.) 分别转化为 T 和 F。
数值型 (包括双精度 型、浮点型或整 型)	从数值型值的小数点部分删除后续零。如果该数值型值是一个整数，则在转换后值中不包含小数点 (例如，4.0 转换为 4)。如果该数值型值小于 1 大于负 1，则在小数点前面包含零 (例如 .4 转换为 0.4)。
通用型	如果通用字段包含一个对象，则返回 "Gen"；否则返回 "gen"。
备注型	没有转换

示例

```
STORE 12.34 TO gnPrice  
CLEAR  
? TRANSFORM ( gnPrice, '$$$$.99')  && 显示 $12.34
```

请参阅

[DTCOC \(\)](#) , [TTOC \(\)](#) , [InputMask](#) 属性

`_TRANSPORT` 系统变量

在不同的 FoxPro 平台和版本之间转换表单、标签和报表时，指定需要运行的程序。包含该变量是为了提供向后兼容性，可使用 Visual FoxPro 转换器代替。

`_TRIGGERLEVEL` 系统变量

包含一个表示当前触发器过程嵌套数目的只读数字值。

语法

```
_TRIGGERLEVEL = nExpression
```

参数描述

nExpression

包含当前触发器过程嵌套数目，当初始的触发器过程执行时，
_TRIGGERLEVEL 为 1；如果一个触发器过程引发了另一个触发器过程，
_TRIGGERLEVEL 就增加 1；如果没有执行触发器过程，_TRIGGERLEVEL
为 0。

说明

使用 CREATE TRIGGER 可以创建一个表的“删除”、“插入”或“更新”触发器。
可以使用 APPEND PROCEDURES 和 MODIFY PROCEDURES 建立每个触发器的内部
存储过程，当触发“删除”、“插入”或“更新”触发器时执行。
_TRIGGERLEVEL 所包含的数字值是只读的。忽略用 STORE 或“=”命令为
_TRIGGERLEVEL 所赋的值。

请参阅

[APPEND PROCEDURES](#), [CREATE TRIGGER](#), [MODIFY PROCEDURE](#)



返回总目录

TRIM () 函数

TTOC () 函数

TTOD () 函数

TXNLEVEL () 函数

TXTWIDTH () 函数

TYPE 命令

Type 属性

TYPE () 函数

TypeLibCLSID 属性

TypeLibDesc 属性

TypeLibName 属性

UIEnable 事件

UndoCheckOut 方法

UnDock 事件

UNIQUE () 函数

_UNIX 系统变量

Unload 事件

UNLOCK 命令

UpClick 事件

UPDATE-SQL 命令

UPDATE 命令

UPDATED()函数

UPPER () 函数

USE 命令

USED () 函数

VAL () 函数

VARTYPE () 函数

Valid 事件

VALIDATEDATABASE 命令

Value 属性

VARREAD () 函数

VERSION () 函数

Version 属性

VersionComments 属性

VersionCompany 属性

VersionCopyright 属性

VersionDescription 属性

VersionLanguage 属性

VersionNumber 属性

VersionProduct 属性

VersionTrademarks 属性

_VFP 系统变量

View 属性

ViewPortHeight 属性

ViewPortLeft 属性

ViewPortTop 属性

ViewPortWidth 属性

Visible 属性

VScrollSmallChange 属性

WAIT 命令

WBORDER () 函数

WCHILD () 函数

WCOLS () 函数

WEEK () 函数

WEXIST () 函数

WFONT () 函数

WhatsThisButton 属性

WhatsThisHelp 属性

WhatsThisHelpID 属性

WhatsThisMode 方法

When 事件

Width 属性

WindowList 属性

WindowState 属性

_WINDOWS 系统变量

WindowType 属性

WITH...ENDWITH 命令

_WIZARD 系统变量

WLAST () 函数

WLCOL () 函数

TRIM () 函数

返回删除全部后缀空格后的指定字符表达式。

语法

TRIM(cExpression)

返回值类型

字符型

参数描述

cExpression

指定的字符表达式， TRIM()函数删除此字符表达式中的全部后缀空格。

说明

TRIM()函数等同于 RTRIM()函数。

示例

```
CLOSEDATABASES  
OPENDATABASE(HOME(2)+'Data\testdata')  
USEcustomer&&打开“Customer”表  
CLEAR  
?'Thecontactfor'+TRIM(company)+'is'+contact
```

请参阅

ALLTRIM(),LTRIM(),RTRIM()

TTOC () 函数

按指定的格式从日期时间表达式中返回一个字符值。

语法

TTOC(*tExpression*[,1|2])

返回值类型

字符型

参数描述

tExpression

指定的一个日期时间表达式，TTOC()函数从该表达式中返回一个字符值。*tExpression*必须是一个合法的日期时间值。如果*tExpression*中只包含时间，VisualFoxPro把默认日期“12/30/1899”添加到*tExpression*中；如果*tExpression*中只包含日期，VisualFoxPro将默认的午夜时间“12:00:00A.M.”添加到*tExpression*中。

1

指定TTOC()函数返回一个字符串。该字符串有14个字符，格式为

“yyyy:mm:dd:hh:mm:ss”，此格式不受 SETCENTURY 或 SETSECONDS 中当前设置的影响。

2

TTOC()返回的字符串只包含日期时间表达式的时间部分。字符串中是否包含秒由 SETSECONDS 和 SETDATE 决定。如果 SETDATE 设为 LONG 或 SHORT，则字符串的格式由“控制面板”中的相应设置决定。

示例

下面的示例创建了一个日期时间型的变量 gtDtime。TYPE()显示 T 就表示此变量是日期时间型。TTOC()用于将变量转换为字符型，现在 TYPE()显示 C，这表示变量已转换为字符型。

```
STOREDATETIME()TOgtDtime&&创建日期时间型变量
```

```
CLEAR
```

```
?"gtDtimeistype:"
```

```
??TYPE('gtDtime')&&显示 T，日期时间型值
```

```
gtDtime=TTOC(gtDtime)&&转换 gtDtime 为字符值
```

```
?"gtDtimeisnowtype:"
```

```
??TYPE('gtDtime')&&显示 C，字符型值
```

请参阅

[DATE\(\)](#),[DATETIME\(\)](#),[HOUR\(\)](#),[MINUTE\(\)](#),[SEC\(\)](#),[SECONDS\(\)](#),[SETCENTURY](#),
[SETDATE](#),[SETSECONDS](#),[TIME\(\)](#),[TTOD\(\)](#)

TTOD () 函数

从日期时间表达式中返回一个日期值。

语法

TTOD(*tExpression*)

返回值类型

日期型

参数描述

tExpression

指定的一个日期时间表达式，TTOD()函数从该表达式中返回一个日期。

tExpression 必须是一个合法的日期时间值。如果 *tExpression* 只包含时间，

VisualFoxPro 将默认的日期“12/30/1899”添加到 *tExpression* 中，并返回这个默认的日期。

示例

下面的示例创建了一个日期时间型的变量 `gtDtime`。TYPE()显示 T 就表示此变量是日期时间型。TTOD()用于将变量转换为日期型，现在 TYPE()显示 D，这表示变量已转换为日期型。

```
STORE DATETIME() TO gtDtime && 创建日期时间型变量
```

```
CLEAR  
?"gtDtimeistype:"  
??TYPE('gtDtime')&&显示 T, 日期时间型值
```

```
gtDtime=TTOD(gtDtime)&&转换 gtDtime 为日期值  
?"gtDtimeisnowtype:"  
??TYPE('gtDtime')&&显示 D, 字符型值
```

请参阅

[DATE\(\)](#),[DATETIME\(\)](#),[HOUR\(\)](#),[MINUTE\(\)](#),[SEC\(\)](#),[SECONDS\(\)](#),[SETCENTURY](#),
[SETSECONDS](#),[TIME\(\)](#),[TTOC\(\)](#)

TXNLEVEL () 函数

返回一个表明当前事务级别的数值。

语法

```
TXNLEVEL()
```

返回值类型

数值型

说明

使用 `BEGINTRANSACTION` 可以创建一个事务。如果在进程中有另一个事务同时使

用 BEGINTRANSACTION 命令，将产生事务的嵌套。事务可按这种方式嵌套 5 层，使用 TXNLEVEL() 函数可确定当前事务的层次。

TXNLEVEL() 函数返回 0 到 5 之间的一个值。如果进程中没有任何事务，TXNLEVEL() 返回 0。

示例

下面的示例打开了数据库 testdata 中的 customer 表。BEGINTRANSACTION 用来开始一个事务处理，TXNLEVEL() 可以显示事务处理级别(1)。BEGINTRANSACTION 还用于开始一个嵌套的事务处理，并且 TXNLEVEL() 为当前的事务处理级别显示 2。

```
CLOSEDATABASES
OPENDATABASE(HOME(2)+'Data\testdata')
USECustomer&&打开“customer”表
CLEAR

BEGINTRANSACTION
  tLevel=ALLTRIM(STR(TXNLEVEL()))
  =MESSAGEBOX("CurrentTransaction:"+tLevel,0,"TrasactionLevel")
  BEGINTRANSACTION
    tLevel=ALLTRIM(STR(TXNLEVEL()))
    =MESSAGEBOX("CurrentTransaction:"+tLevel,0,;
      "TrasactionLevel")
  ENDTRANSACTION
ENDTRANSACTION
```

请参阅

[BEGINTRANSACTION,ENDTRANSACTION,ROLLBACK](#)

TXTWIDTH () 函数

返回有关字体的字符表达式的长度。

语法

TXTWIDTH(*cExpression* [, *cFontName*, *nFontSize* [, *cFontStyle*]])

返回值类型

数值型

参数描述

cExpression

指定需要 TXTWIDTH() 函数返回其长度的字符表达式。如果只有 *cExpression* 参数，则按 VisualFoxPro 主窗口或活动输出窗口中的当前字体返回字符表达式长度。使用非比例字体总是返回与 *cExpression* 参数中字符数目相等的一个数值。

包含可选参数 *cFontName*，*nFontSize* 和 *cFontStyle*，可以确定具有特殊的字体、磅值和字形的字符表达式长度。

cFontName

指定字体的名称。包含 *cFontName* 参数可以返回特殊字体的字符表达式长度。

nFontSize

指定一个对应于字体磅值的数值。包含 *nFontSize* 参数可以返回具有特殊磅值的字符表达式长度。

cFontStyle

指定字符表达式的特定字形。*cFontStyle* 是由一个字符或一组字符指定的字形代码。如果不包含 *cFontStyle* 参数，则使用常规字形。

下面是每种字形的字符代码：

字符

字形

B	粗体
I	斜体
N	常规
O	轮廓
Q	不透明
S	阴影
-	删除线
T	透明
U	下划线

可以使用多个字符指定字形的组合。例如，“BI”指定粗斜体。

下面的命令用“Courier”普通24点阵字体，在VisualFoxPro主窗口水平居中地显示一段信息：

```
@25,(WCOLS()-TXTWIDTH('Hello!','Courier',24)*;  
    FONTMETRIC(6,'Courier',24)/FONTMETRIC(6))/2;  
    SAY'Hello!'FONT'Courier',24
```

说明

TXTWIDTH()函数按照指定字体的平均字符宽度（字体的平均字符宽度称为“foxel”），返回字符表达式 *cExpression* 中包含的字符数目。使用 FONTMETRIC(6) 可确定字体平均字符宽度。

如果在 TXTWIDTH() 中不包含字体参数，并且在调用 TXTWIDTH() 函数时 VisualFoxPro 主窗口是输出窗口，TXTWIDTH() 函数的返回值就由 VisualFoxPro 主窗口中的当前字体决定。如果输出到用户自定义窗口中，返回值就由用户自定义窗口的当前字体决定。

请参阅

[FONTMETRIC\(\)](#), [GETFONT\(\)](#), [LEN\(\)](#), [SYSMETRIC\(\)](#), [WFONT\(\)](#)

TYPE 命令

显示文件的内容。

语法

TYPE FileName1

[AUTO]

[WRAP]

[TOPRINTER[PROMPT]|TOFILE FileName2]

[NUMBER]

参数描述

FileName1

指定文件的名称，该文件名必须包含扩展名。

AUTO

自动缩进。当包含 WRAP 时，对于每个段落中换行的文本，TYPE 自动缩进与段落第一行相等的距离。例如：

- 当段落以一个制表位开始，换行的文本与该制表位的位置对齐。
- 当段落以数字（或其他文本）开始，其后接制表位和文本时，换行的文本与该制表位的位置对齐
- 当段落的第一行以空格缩进时，每一换行的文本都缩进与第一行缩进相同的距离。

WRAP

允许自动换行，这样，不能放在行末尾的太长的单词将自动移到下一行。

TOPRINTER[PROMPT]

输出到打印机。

在 VisualFoxPro 中，可以加入可选的 PROMPT 子句，在开始打印时显示一个打印对话

框。PROMPT 关键字应该直接放在 TOPRINTER 后面。

TOFILEFileName2

输出到文件 *FileName2* 中。

NUMBER

在输出中的每行开始加一个行号。

说明

TYPE 命令显示文件的内容。这些显示内容可以输出到 VisualFoxPro 主窗口、活动的用户自定义窗口、打印机或另一个文件中。

当 SETHEADINGS 为 ON 时，VisualFoxPro 在 TYPE 命令输出的开头插入一个换页符、文件的路径名称以及日期；如果 SETHEADINGS 为 OFF，则不包含以上内容。

在 MS-DOS 版本的 FoxPro 中，如果上载了打印机驱动程序设置，并且您用 TYPE 命令从文件或打印机上得到输出，TYPE 显示使用打印机驱动程序安装中设置的文件内容。

请参阅

[SETHEADINGS](#)

Type 属性

包含一个字符，表明项目中一个文件的文件类型。设计和运行时可用。

语法

Object.Type

说明

下表列出了 Type 属性可以包含的值，以及相应的文件类型。

值	FoxPro.H 常数	文件类型和扩展名
d	FILETYPE_DATABASE	数据库， .dbc
D	FILETYPE_FREETABLE	自由表， .dbf
Q	FILETYPE_QUERY	查询， .qpr
K	FILETYPE_FORM	表单， .scx
R	FILETYPE_REPORT	报表， .frx
B	FILETYPE_LABEL	标签， .lbx
V	FILETYPE_CLASSLIB	可视类库， .vcx
P	FILETYPE_PROGRAM	程序， .prg
L	FILETYPE_APILIB	VisualFoxPro 动态链接库， .fl1
Z	FILETYPE_APPLICATION	应用程序， .app
M	FILETYPE_MENU	菜单， .mnx
T	FILETYPE_TEXT	文本文件， 可变
x	FILETYPE_OTHER	其他， 可变

这些值保存在项目 .pjx 文件中的 Type 字段。

应用于

文件对象

请参阅

[FileClass 属性](#), [FileClassLibrary 属性](#)

TYPE () 函数

计算字符表达式，并返回表达式的数据类型。

语法

TYPE(cExpression)

返回值类型

字符型

参数描述

cExpression

指定用来评估的表达式，它可以是变量、字段、备注字段或其他任意表达式。表达式必须作为字符串传递；用引号将变量、字段等的名称引起来。如果表达式没有用引号，TYPE()函数将返回“U”（未定义的表达式）。

说明

下表列出了 TYPE()函数所返回的字符值及其对应的数据类型：

数据类型

返回的字符

字符型	C
数值型 (或者整数、单精度浮点 数和双精度浮点数)	N
货币型	Y
日期型	D
日期时间型	T
逻辑型	L
备注型	M
对象型	O
通用型	G
屏幕 (使用 SAVESCREEN 创建的)	S
未定义的表达式类型	U

示例

```
CLOSEDATABASES  
OPENDATABASE(HOME(2)+'Data\testdata')  
USEcustomer&&打开“Customer”表  
nTest=1.01  
cTest="String"
```

```
CLEAR
?TYPE ('customer.contact')&&显示 C
?TYPE ('(12*3)+4')&&显示 N
?TYPE ('DATE()')&&显示 D
?TYPE ('.F.OR.T.')&&显示 L
?TYPE ('ANSWER=42')&&显示 U
?TYPE ('$19.99')&&显示 Y
?TYPE ('nTest')&&显示 N
?TYPE ('cTest')&&显示 C
```

请参阅

[EVALUATE\(\)](#)

TypeLibCLSID 属性

为项目中服务程序类创建的类型库的注册 CLSID。只读。

语法

Object.TypeLibCLSID

说明

当根据包含有服务程序类的项目连编一个 .dll 或 .exe 文件时，会创建和注册一个类型库。类型库的扩展名为 .tlb，并且放在与 .dll 或 .exe 文件相同的目录中。

应用于

项目对象

请参阅

[CLSID 属性](#), [ProgID 属性](#), [TypeLibDesc 属性](#), [TypeLibName 属性](#)

TypeLibDesc 属性

为项目中服务程序类创建的类型库的说明。

语法

```
Object.TypeLibDesc[=cTypeLibraryDescription]
```

参数描述

`cTypeLibraryDescription`

指定为项目中服务程序类创建的类型库的说明。在默认情况下，包含项目名称，后面加“TypeLibrary”。

说明

当根据包含有服务程序类的项目连编一个 .dll 或 .exe 文件时，会创建和注册一个类型库。类型库的扩展名为 .tlb，并且放在与 .dll 或 .exe 文件相同的目录中。

该属性对应于“项目信息”对话框“服务程序”选项卡的“类型库说明”项。

应用于

项目对象

请参阅

[CLSID 属性](#), [ProgID 属性](#), [TypeLibCLSID 属性](#), [TypeLibName 属性](#)

TypeLibName 属性

为项目中服务程序类创建的类型库的名称。只读。

语法

Object.TypeLibName

说明

当根据包含有服务程序类的项目连编一个 .dll 或 .exe 文件时，会创建和注册一个类型库。类型库的扩展名为 .tlb，并且放在与 .dll 或 .exe 文件相同的目录中。

应用于

项目对象

请参阅

[CLSID 属性](#), [Name 属性](#), [ProgID 属性](#), [TypeLibCLSID 属性](#), [TypeLibDesc 属性](#)

UIEnable 事件

无论页面是否激活，包含在该页面内的所有对象都能发生 UIEnable 事件。

语法

```
PROCEDURE Object.UIEnable  
[LPARAMETERS nIndex, IEnable]
```

-或者-

```
PROCEDURE Object.UIEnable  
LPARAMETERS IEnable
```

参数描述

nIndex

存放一个编号，该编号标识控件数组中的一个控件。只有当一个控件在一个控件数组中时才传递 *nIndex* 参数。

IEnable

存放一个逻辑值，该逻辑值指定包含对象的页面是否激活。如果 *IEnable* 是“真” (.T.)，则包含对象的页面是激活的（成为活动页面），如果 *IEnable* 是“假” (.F.)，该页面是非激活的（成为非活动的页面）。

说明

使用 UIEnable 事件可以指定在激活页面或使页面不活动时，包含在页面中的对象或控件发生的行为。

当表单初始激活时，页面对象的 UIEnable 事件并不发生。只有当页面是以编程方式或交互方式激活或变成不活动时，才能发生 UIEnable 事件。

应用于

复选框，组合框，命令按钮，命令组，容器对象，控件对象，编辑框，表格，图像，标签，线条，列表框，OLE 绑定型控件，OLE 容器控件，选项组，页框，形状，微调，文本框

请参阅

[Activate 事件](#), [Deactivate 事件](#)

UndoCheckOut 方法

放弃对文件的所有更改，并且将该文件签回源代码管理器。

语法

```
Object.UndoCheckOut()
```

说明

如果成功地放弃了对文件的所有更改，并且该文件签回了源代码管理器，则返回“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](#), [RemoveFromSCC 方法](#)

UnDock 事件

从停放位置拖动工具栏时发生。

语法

```
PROCEDUREToolBar.Undock  
[LPARAMETERS nIndex]
```

参数描述

nIndex

用以唯一标识控件数组中的一个控件。

应用于

工具栏

请参阅

[AfterDock 事件](#), [BeforeDock 事件](#), [Dock 方法](#)

UNIQUE () 函数

包含此函数是为了提供与 dBASE 的兼容性。

_UNIX 系统变量

在使用 FoxProforUNIX 时为“真”(.T.)。

语法

`_UNIX=IExpression`

说明

如果正在使用 FoxProforUNIX，_UNIX 系统变量为“真”(.T.)。如果使用的不是 FoxProforUNIX 版本，_UNIX 变量为“假”(.F.)。

包含在 _UNIX 变量中的值不能用 STORE 或 = 命令改变。

请参阅

[_DOS,_MAC,VERSION\(\),_WINDOWS](#)

Unload 事件

在对象被释放时发生。

语法

```
PROCEDURE Object.Unload  
[LPARAMETERS nIndex]
```

参数描述

nIndex

用以唯一标识控件数组中的一个控件。

说明

Unload 事件是在释放表单集或表单之前发生的最后一个事件。Unload 事件发生在 Destroy 事件和所有包含的对象被释放之后。

这一事件的发生取决于对象的类型：

- 当释放引用表单的对象变量或该表单的表单集时，表单对象以代码形式释

放。

- 当释放引用表单集的对象变量时，表单集对象以代码形式释放。
如果一个容器对象，例如表单集，包含多个对象，则该容器对象的 Unload 事件发生在其所包含的对象的 Unload 事件之后。例如，一个表单集中包含一个表单，该表单中包含一个控件（一个命令按钮），释放的顺序如下：
 1. 表单集的 Destroy 事件
 2. 表单的 Destroy 事件
 3. 命令按钮的 Destroy 事件
 4. 表单的 Unload 事件
 5. 表单集的 Unload 事件

应用于

表单，表单集

请参阅

[Destroy 事件](#)

UNLOCK 命令

对一个表中的单条记录、多条记录或者文件解锁，或者对所有打开的表解除所有记录锁和文件锁。

语法

UNLOCK

[RECORD *nRecordNumber*]

[IN *nWorkArea*|*cTableAlias*]

[ALL]

参数描述

RECORD nRecordNumber

解除第 *nRecordNumber* 个记录上的记录锁。可以令 *nRecordNumber* 为 0，来解除 LOCK(0)或 RLOCK(0)锁定的表头。

解除第 *nRecordNumber* 个记录上的记录锁。可以令 *nRecordNumber* 为 0，来解除 LOCK(0)或 RLOCK(0)锁定的表头。

IN nWorkArea|cTableAlias

从指定工作区内的表中解除记录锁或文件锁。*nWorkArea* 指定工作区号，而

cTableAlias 指定表的别名。如果不包含 *nWorkArea* 或 *cTableAlias* 参数，则 UNLOCK 从当前工作区内的表中解除记录锁或文件锁。

ALL

解除所有工作区中的全部记录锁和文件锁。

有关网络上的记录、文件锁定与共享表的详细内容，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”。

说明

记录锁和文件锁只能由加锁的用户从表中删除。UNLOCK 命令不能解除独占使用的表的锁定。

如果执行不带参数的 UNLOCK 命令，则解除当前工作区中表的记录锁或文件锁。

如果执行不带参数的 UNLOCK 命令，则解除当前工作区中表的记录锁或文件锁。

示例

```
CLOSEDATABASES
SETREPROCESSTO3AUTOMATIC
STORE'1,2,3,4'TOgcRecList&&锁定记录 1 至 4
gcOldExc=SET('EXCLUSIVE')&&保存“EXCLUSIVE”设置
SETEXCLUSIVEOFF&&允许表共享
```

```
CLOSEDATABASES
OPENDATABASE(HOME(2)+'Data\testdata')
USEcustomer&&打开“Customer”表
SELECT0
USEemployee&&打开“Employee”表
```

```
?LOCK('1,2,3,4','customer')&&锁定“customer”中的头四个记录
?RLOCK(gcRecList,'employee')&&锁定“employee”中的头四个记录 employee
```

```
UNLOCKIN customer
UNLOCKIN employee
SETEXCLUSIVE&gcOldExc&&保存原始的“EXCLUSIVE”设置
```

请参阅

[FLOCK\(\),LOCK\(\),RLOCK\(\)](#)

UpClick 事件

在用户单击控件的向上滚动箭头时发生。

语法

```
PROCEDURE Control.UpClick  
[LPARAMETERS nIndex]
```

参数描述

nIndex

用以唯一标识控件数组中的某个控件。

应用于

组合框，微调

请参阅

[DownClick 事件](#)

UPDATE-SQL 命令

以新值更新表中的记录。

语法

```
UPDATE[DatabaseName1!]TableName1  
SETColumn_Name1=eExpression1  
[,Column_Name2=eExpression2...]  
WHEREFilterCondition1[AND|ORFilterCondition2...]]
```

参数描述

[DatabaseName1!]TableName1

TableName1 指定要更新记录的表。

DatabaseName1 指定包含表的非当前数据库名。如果包含表的数据库不是当前数据库，则应包含这个数据库名。在数据库名称与表名之间有一个感叹号(!)。

SETColumn_Name1=eExpression1

[,Column_Name2=eExpression2

指定要更新的列以及这些列的新值。如果省略了 WHERE 子句，在列中的每

一行都用相同的值更新。

`WHERE FilterCondition1[AND|OR FilterCondition2...]`
指定要更新的记录。

FilterCondition 指定要更新的记录所符合的条件。可以根据需要加入多个筛选条件，条件之间用 AND 或 OR 操作符连接。也可以用 NOT 操作符对逻辑表达式的值取反，或者使用 EMPTY() 函数检查字段是否为空。

说明

UPDATE-SQL 命令只能用来更新单个表中的记录。注意，UPDATE-SQL 支持子查询。

与 REPLACE 命令不同的是，在更新可共享访问的表中的多个记录时，UPDATE-SQL 命令使用了记录锁。在多用户的情况下，使用记录锁能减少记录竞争，但是可能会降低使用性能。为了达到最好的性能，请以独占使用方式打开表或用 FLOCK() 函数锁定表。

示例

下面的示例打开了数据库 testdata 中的 customer 表。UPDATE-SQL 用于将 maxordamt 字段的所有值设置为 25。

```
CLOSEDATABASES
```

```
OPENDATABASE(HOME(2)+'Data\testdata')
```

```
USECustomer&&打开“customer”表
```

```
*Setanddisplayamountsforcustomers
```

```
UPDATEcustomerSETmaxordamt=25
```

BROWSEFIELDScompany,maxordamt

请参阅

[DELETE-SQL](#),[GATHER](#),[INSERT-SQL](#),[REPLACE](#),[SCATTER](#)

UPDATE 命令

包含此命令是为了提供向后兼容性。请使用 [UPDATE-SQL](#) 代替。

UPDATED()函数

包含此函数是为了提供向后兼容性。请使用 `InteractiveChange` 或 `ProgrammaticChange` 事件。

UPPER () 函数

用大写字母返回指定的字符表达式。

语法

UPPER(cExpression)

返回值类型

字符型

参数描述

cExpression

指定字符表达式，UPPER()函数将它转换成大写字母。

说明

在字符表达式中的每一个小写字母(a-z)在返回串中都将转换成大写字母，其他字符保持不变。

示例

```
CLEAR  
?UPPER('abcdefgh')&&显示 “ABCDEFGH”
```

请参阅

ISLOWER(),ISUPPER(),LOWER()

USE 命令

打开一个表及其相关索引文件，或打开一个 SQL 视图。

语法

```
USE[[DatabaseName!]Table|SQLViewName|?]  
  [INnWorkArea|cTableAlias]  
  [ONLINE]  
  [ADMIN]  
  [AGAIN]  
  [NOREQUERY[nDataSessionNumber]]  
  [NODATA]  
  [INDEXIndexFileList|?  
  [ORDER[nIndexNumber|IDXFileName  
  |[TAG]TagName[OFCDXFileName]  
    [ASCENDING|DESCENDING]]]  
  [ALIAScTableAlias]  
  [EXCLUSIVE]
```

[SHARED]

[NOUPDATE]

参数描述

[DatabaseName!]TableName

指定要打开的表的名称。尽管在 Windows95 和 WindowsNT 中，文件名内的空格是有意义的，但要避免在 VisualFoxPro 中的 *TableName* 内使用空格符。如果表名中含有空格，请用括号（“ ”或 ‘ ’）括住表名。

要想打开一个不在当前数据库中的表，在表名前加数据库的名称，中间用叹号分隔。如果不包含数据库的名称，MicrosoftVisualFoxPro 只能在当前数据库中打开表。如果数据库名或表名或者两者中都含有空格，请用括号（“ ”或 ‘ ’）括住数据库名和表名。

SQLViewName

指定位于当前数据库内要打开的 SQL 视图的名称。SQL 视图可以用 `CREATE SQLVIEW` 命令创建。SQLViewName 可作为使用 `CREATE OFFLINE()` 创建的游离视图的名称。

?

显示“使用”对话框，可以从对话框中选择要打开的表。

INnWorkArea

指定要打开表所在的工作区。带有 IN 子句和工作区编号的 USE 命令，可以关闭指定工作区中的表。

在 IN 子句中 0 可作为工作区号，指定 0 可以在最低可用的工作区中打开表。例如，如果工作区 1 至工作区 10 中都有表打开，那么下面的命令将在工作区 11 中打开

customer 表：
USEcustomerIN0

INcTableAlias

指定在当前工作区中打开表。打开表的别名用 *cTableAlias* 指定。
如果省略 *nWorkArea* 和 *cTableAlias* 参数，则在当前工作区中打开表。

ONLINE

打开一个游离视图 (offlineview)。有关游离视图的创建，请参见
CREATEOFFLINE()。要打开的游离视图的名称由 *SQLViewName* 指定。使用
ONLINE 打开游离视图后，您可用 TABLEUPDATE() 将它的改变更新到服务器中。

使用 ONLINE 打开游离视图只能采用独占方式。所以应在 USE 命令中同时使用
EXCLUSIVE 子句，或在 USE 命令前使用 SETEXCLUSIVEON。

ADMIN

打开一个游离视图，而且打开的目的是为了操作其中的数据，而不是将游离
视图中的更改更新到服务器中。有关游离视图的创建，请参见
CREATEOFFLINE()。使用 ADMIN 打开游离视图后，可以对它进行修改而
不会影响服务器中的数据。

AGAIN

若要同时在多个工作区中打开一个表，可以按以下方法操作：

- 选择另一个工作区，并且执行带有表名和 AGAIN 子句的 USE 命令。
- 执行带有表名和 AGAIN 子句的 USE 命令，并且用 IN 子句指定一个不同

的工作区。

当在另一个工作区中再次打开一个表时，在新工作区中的表继承了原工作区中的表的属性。例如，如果一个表以只读或独占访问方式打开，然后在另一个工作区中打开，那么在新的工作区中该表也以只读或独占访问方式打开。

当再次打开表时，如果没有打开索引，则使用为原表打开的索引文件。在再次打开该表的工作区中，设置索引序号为 0。

即使初始表没有打开索引，也可以在再次打开该表时打开索引。这将把初始表的索引序号设置成 0。

再次打开的表被赋予工作区的默认别名。在多个工作区中打开一个表时，每次可以包含一个别名，只要这些别名不重复就可以。

在 VisualFoxPro 中，在另一个工作区中再次打开某个表并不占用额外的文件句柄。

NOREQUERY[nDataSessionNumber]

指定远程 SQL 视图的数据不再下载。NOREQUERY 仅用于 SQL 视图，并且主要用在当 AGAIN 子句再次打开一个 SQL 视图的时候。包含 NOREQUERY 子句可以增加大型数据集合的操作性能，因为数据不需要再次下载。

nDataSessionNumber 可以指定在某一数据工作期中的远程 SQL 视图内的数据不被再次下载。如果省略了 *nDataSessionNumber*，则不下载在当前数据工作期中打开的视图。有关 NOREQUERY 子句的详细内容，请参阅《Microsoft VisualFoxPro 6.0 中文版程序员指南》的第八章“创建视图”。

NODATA

指定仅下载 SQL 视图的结构，不下载 SQL 视图的数据。NODATA 提供了确定 SQL 视图结构的最快捷方法。

有关 NODATA 子句的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第八章“创建视图”。

INDEX IndexFileList

指定一组和表一起打开的索引。如果表具有结构复合索引文件，该索引文件自动与表一起打开。

IndexFileList 可以包含任何 .IDX 单项索引文件和 .CDX 复合索引文件的文件名。除非在索引文件列表中的 .IDX 和 .CDX 索引文件具有相同的文件名，否则无须为索引文件加扩展名。

在索引文件列表中的第一个索引文件是主控索引文件，该文件控制表中的记录如何访问和显示。然而，如果第一个索引文件是一个 .CDX 复合索引文件，则表中的记录按记录的物理顺序显示和访问。

INDEX?

显示“打开”对话框，列出所有可供选择的索引文件。

ORDER[nIndexNumber]

指定一个主控的 .IDX 单项索引文件或 .CDX 复合索引文件标识，而不是使用在 *IndexFileList* 中指定的第一个索引文件或索引标识作为主控索引。

索引编号按如下方式：首先编号 .IDX 索引文件，编号顺序是其出现在索引文件列表中的顺序。然后对结构复合索引文件（如果存在）中的标识按其创建顺序编号。最后，对任何独立的复合索引文件的标识按其创建顺序编号。也可以使用 SETORDER 来指定

主控索引文件或标识。有关索引文件和标识编号的详细内容，请参阅 SETORDER。如果 *nIndexNumber* 为 0，表中的记录将按物理记录顺序显示和访问，且各索引继续保持打开状态。尽管包含 ORDER0 将以记录号顺序展现文件，但仍允许对已打开索引文件的更新。不加 *nIndexNumber* 的 ORDER 子句与 ORDER0 意义相同。

ORDER[IDXFileName]

指定一个 .IDX 单项索引文件作为主控索引文件。

ORDER[TAGTagName][OFCDXFileName]

指定 .CDX 复合索引文件中的主控标识。标识名可以是来自结构复合索引文件或任何其他打开的复合索引文件。如果在已打开的各复合索引文件中存在相同的标识名，则应包含 *OFCDXFileName*，并且指定复合索引文件名。

ASCENDING

指定表中的记录按升序访问和显示。

DESCENDING

指定表中的记录按降序访问和显示。

包含 ASCENDING 或 DESCENDING 子句并不改变索引文件或标识，而仅仅改变记录显示和访问的顺序。

ALIAScTableAlias

创建表的别名。可以在需要或支持别名的命令和函数中用别名来引用表。

打开表时，系统自动给它指定一个别名。如果不含 ALIAS 子句，那么就用该表的名称作为表的别名，但也可以包含 ALIAS 子句和一个新别名来为表创建一个不同的别名。

在 VisualFoxPro 中，别名可以包含多达 254 个字母、数字或下划线，并且必须以字母或

下划线开头。在其他 FoxPro 版本中，别名最多可包含 10 个字母、数字或下划线，并且必须以字母或下划线开头。

如果使用 AGAIN 子句在多个工作区中同时打开一个表，并且在每个工作区中打开该表时不指定别名，那么每次将自动赋给该表一个默认的别名。

指定的别名发生冲突时，也赋给该表一个默认的别名，例如：

```
CLOSEDATABASES
```

```
OPENDATABASE(HOME(2)+'Data\testdata')
```

```
ACTIVATEWINDOW View&&打开查看窗口
```

```
USEcustomerALIASordersIN1&&别名是 ORDERS
```

```
USEordersIN3&&冲突，别名是 c
```

EXCLUSIVE

在网络上以独占使用方式打开表。有关表的独占使用的详细内容，请参阅 SETEXCLUSIVE。

SHARED

在网络上以共享使用方式打开表。用 SHARED 子句打开一个表时，即使 EXCLUSIVE 设置成 ON，此表也将以共享方式使用。

NOUPDATE

禁止更改表及其结构。

说明

如果不加表名执行 USE 命令，并且在当前工作区中已经打开了一个表文件，则关闭这个表。同样，在一个工作区中打开另一个表时，原来在该工作区中打开的表自动关

闭。不能在同一个工作区中同时打开多个表。

示例

下面的示例在三个不同的工作区中打开了三个表。“数据工作期”窗口也被打开以显示打开表的地点和每个表的别名。

```
CLOSEDATABASES  
OPENDATABASE(HOME(2)+'Data\testdata')  
ACTIVATEWINDOWView
```

```
USEcustomerIN0&&打开“Customer”表  
USEemployeeIN0&&打开“Employee”表  
USEproductsIN0&&打开“Products”表
```

请参阅

[CREATE,CREATESQLVIEW,CREATETABLE-SQL,CREATEOFFLINE\(\),DBF\(\),INDEX,USED\(\)](#)

USED () 函数

确定是否在使用一个别名，或者是否在指定工作区中打开了一个表。

语法

USED([nWorkArea|cTableAlias])

返回值类型

逻辑型

参数描述

nWorkArea|cTableAlias

指定表的工作区或别名。如果表在 nWorkArea 指定的工作区中打开，USED() 将返回逻辑真(.T.)；否则返回逻辑假(.F.)。如果 cTableAlias 指定的别名正在使用，USED()将返回逻辑真(.T.)；否则返回逻辑假(.F.)。

如果省略了 nWorkArea 与 cTableAlias 并且在当前选定的工作中打开表，USED()将返回逻辑真(.T.)；否则返回逻辑假(.F.)。

说明

USED()可以判断一个别名是否正在使用中，或者是否在指定的工作区中打开了一个表。

示例

```
CLOSEDATABASES
OPENDATABASE(HOME(2)+'Data\testdata')
SELECTA
USEcustomer&&打开“Customer”表
SELECTB
USEorders&&打开 Order 表
SELECTC
USEemployee&&打开 Employee 表
?USED('A')&&显示.T.
?USED('B')&&显示.T.
?USED(4)&&显示.F.
```

请参阅

[ALIAS\(\)](#),[SELECT\(\)](#)

VAL () 函数

将数字组成的字符表达式转换成数字值。

语法

VAL(cExpression)

返回值类型

数值型

参数描述

cExpression

指定字符表达式，该表达式由最多由 16 位的数字组成，若超过 16 位，则对其圆整。

说明

VAL()函数从左到右返回字符表达式中的数字，直至遇到非数值型字符（忽略前面的

空格) 时为止。若字符表达式的第一个字符不是数字, 也不是正、负号, 则 VAL() 函数返回 0。

VAL() 函数可以将 VisualFoxPro 的 SYS() 函数返回的字符串转换为数值。

示例

```
CLEAR  
STORE '12' TO A  
STORE '13' TO B  
? VAL(A)+VAL(B) && 显示 25.00  
STORE '1.25E3' TO C  
? 2*VAL(C) && 显示 2500.00
```

请参阅

[SETDECIMALS,STR\(\)](#)

VARTYPE () 函数

返回一个表达式的数据类型。

语法

```
VARTYPE (eExpression [, INullDataType])
```

返回值类型

字符型

参数描述

eExpression

指定要返回数据类型的表达式。VARTYPE()返回单个的字符，表明该表达式的数据类型。下表列出了 VARTYPE()返回的每种数据类型对应的字符：

返回的字符

数据类型

C	字符型或备注型
N	数值型、整型、浮点型或双精度型
Y	货币型
L	逻辑型
O	对象
G	通用型
D	日期型
T	日期时间型
X	Null
U	未知

注意如果 eExpression 是一个数组，则返回该数组第一个值的数值类型。

INullDataType

指定当 eExpression 包含 null 值时，VARTYPE()是否返回它的数据类型。如果 INullDataType 为“真”(.T.)，则 VARTYPE()返回 eExpression 的数据类型。如果 INullDataType 为“假”(.F.)或省略，则 VARTYPE()返回 'X'，表明

eExpression 包含 null 值。

说明

VARTYPE()类似于 TYPE()函数，但是 VARTYPE()更快，而且表达式外面不需要引号。

如果指定的变量、字段、对象或对象属性不存在，VARTYPE()将返回“U”。如果当前没有活动的表单，运行 ActiveForm 属性会产生 OLEIDispatch 错误。

请参阅

[EVALUATE\(\),TYPE\(\)](#)

Valid 事件

在控件失去焦点之前发生。

语法

```
PROCEDUREControl.Valid  
[LPARAMETERSnIndex]
```

参数描述

nIndex

在控件数组中标识单个控件。

说明

若 Valid 事件返回“真”(.T.)，表明控件失去了焦点；若返回“假”(.F.)，则说明控件没有失去焦点。

Valid 事件也可以返回数值，对应于以下情况：

- 若返回 0，则控件没有失去焦点。
- 若返回正值，则该值指定焦点向前移动的控件数。例如，若 Valid 事件返回 1，则焦点由下一个控件得到。
- 若返回负值，则该值指定焦点向后移动的控件数。例如，若 Valid 事件返回 -1，则焦点由上一个控件得到。

应用于

复选框，组合框，命令按钮，命令组，编辑框，表格，列表框，选项按钮，选项组，微调，文本框

请参阅

[LostFocus 事件](#) , [When 事件](#)

VALIDATEDATABASE 命令

保证当前数据库中表和索引位置的正确性。

语法

```
VALIDATEDATABASE  
  [RECOVER]  
  [NOCONSOLE]  
  [TOPRINTER[PROMPT]|TOFILEFileName]
```

参数描述

RECOVER

显示一个对话框，该对话框允许您定位表和索引，这些表和索引不在被检查的数据库中。必须在命令窗口中发出 `VALIDATEDATABASE RECOVER` 命令，在程序中发布该命令会产生错误信息。

NOCONSOLE

不向 VisualFoxPro 主窗口或活动的用户自定义窗口输出错误信息。

TOPRINTER[PROMPT]

将 `VALIDATEDATABASE` 命令的错误信息定向输出到打印机。

`PROMPT` 在打印前显示“打印”对话框，该关键字应紧跟在 `TOPRINTER` 之后。

TOFILEFileName

将错误信息定向输出到由 *FileName* 参数指定的文件，如果该文件已经存在，并且 `SETSAFETY` 值为 `ON`，系统会询问是否要覆盖该文件。

说明

`VALIDATEDATABASE` 命令确保数据库包含的表和索引处于正确位置，确保数据库中的表包含正确的字段，以及确定数据库中索引标识是否存在。

VALIDATEDATABASE 命令在当前数据库上操作。在发布 OPENDATABASE 命令打开此数据库时，必须包含 EXCLUSIVE 关键字，以独占方式打开。

示例

下面的示例打开 testdata 数据库，并使用 VALIDATEDATABASE 命令，以确保表和索引的位置在数据库中是正确的。

```
CLOSEDATABASES
SETPATHTO(HOME(2)+'Data\')&&设置数据库路径
OPENDATABASEtestdataEXCLUSIVE&&打开 testdata 数据库

VALIDATEDATABASE
```

请参阅

[CREATEDATABASE,MODIFYDATABASE,OPENDATABASE,USE](#)

Value 属性

指定控件的当前状态。设计和运行时可用。对于组合框和列表框控件，此属性只读。

语法

```
[Form.]Control.Value[=nSetting]
```

参数描述

nSetting

对于复选框控件，Value 属性的设置有：

设置	说明
0	（默认值）未选定。
1	选定
2	灰色，该设置只在代码中可用。

对于命令组、组合框、编辑框、列表框、选项组和微调控件，Value 属性的设置为当前所选的字符或数值。

对于选项按钮控件，Value 属性的设置有：

设置	说明
0	表明已选中按钮。
1	（默认值）表明没有选中按钮。

对于文本框控件，Value 属性可以是当前选定的字符串、数值、日期型值、日期时间型值、货币型值或逻辑型值，默认设置为字符串。

说明

对于表格控件，Value 属性只在表格有焦点时可用。

命令组或选项组控件的 Value 属性用来确定组中引起事件的按钮。它被设置为整数，以表明组中引起事件的按钮。

当为控件设置了控件源时，Value 属性的行为将有所改变。在设置控件源时，ControlSource 属性用以指定变量或字段，而 Value 属性则设置该变量或字段的数据类

型。如果此数据类型对给定的控件无效，则 VisualFoxPro 会产生错误。当为控件设置了控件源时，Value 属性的行为将有所改变。在设置控件源时，ControlSource 属性用以指定变量或字段，而 Value 属性则设置该变量或字段的数据类型。如果此数据类型对给定的控件无效，则 VisualFoxPro 会产生错误。

以下为有效数据类型的列表：

控件	所允许的数据类型
复选框	整型、逻辑型、数值型
组合框	字符型、整型、数值型
命令组	字符型、整型、数值型
编辑框	字符型、备注型
表格	字符型、数值型
列表框	字符型、整型、数值型
选项按钮	整型、逻辑型、数值型
选项组	字符型、整型、数值型
微调	货币、整型、数值型
文本框	任何数据类型

应用于

复选框，组合框，命令组，编辑框，表格，列表框，选项按钮，选项组，微调，

文本框

请参阅

[Bound 属性](#), [ListIndex 属性](#)

VARREAD () 函数

包含此项是为了在 VisualFoxPro 中提供向后兼容性。请使用 [ControlSource](#) 或 [Name](#) 属性。

VERSION () 函数

返回一个字符串，该字符串包含了正在使用的 VisualFoxPro 版本号。

语法

VERSION(nExpression)

返回值类型

字符型,数字型

参数描述

nExpression

指定 VERSION() 函数返回有关 VisualFoxPro 的附加信息。若省略 *nExpression* 参数，则 VERSION() 函数返回 VisualFoxPro 的版本号。下表列出了 *nExpression* 参数及返回的有关 VisualFoxPro 的附加信息。

nExpression 返回的有关 VisualFoxPro 的附加信息

- | | |
|---|--|
| 1 | VisualFoxPro 的日期及系列号。 |
| 2 | VisualFoxPro 的版本类型：
0-运行版
1-标准版（早期版本）
2-专业版（早期版本） |
| 3 | 指出 VisualFoxPro 使用的语言，以下两个字符值表明 VisualFoxPro 所使用的语言：
00-英语
33-法语
34-西班牙语
39-意大利语
42-捷克语
48-波兰语
49-德语
55-葡萄牙语
82-朝鲜语
86-中文简体
88-中文繁体 |

- 4 易于分析的标准格式 VisualFoxPro 版本号。标准格式是 "MM.00.YDDD.mm"，其中 MM 是主发布号，00 是固定占位符，YDDD 是四数字的创建该版本的生产日期，mm 是可增加的次级修订编号。
对于生产日期 YDDD，Y 是九十世纪的年数，DDD 是该年的天数。例如，8001 对应于 1998 年 1 月 1 日。
- 5 "Mmm"格式的 VisualFoxPro 发布版本，其中 M 是主发布号，mm 是可增加的次级修订编号。例如，VERSION(5)在 VisualFoxPro6.0 中返回 "600"。

说明

可使用 VERSION()函数有条件地执行与特定版本相关的代码。

VERSION()、VERSION(1)、VERSION(3)和 VERSION(4)返回字符串；VERSION(2)返回一个数值。

示例

```
CLEAR
?VERSION()
?VERSION(1)
?VERSION(2)
?VERSION(3)
?VERSION(4)
?VERSION(5)
```

请参阅

[GETCP\(\)](#), [GETENV\(\)](#), [OS\(\)](#)

Version 属性

将 VisualFoxPro 的一个实例的版本号返回为一个字符串。运行时只读。

语法

`ApplicationObject.Version`

应用于

Application 对象，_VFP 系统变量

请参阅

[VERSION\(\)](#)

VersionComments 属性

一个项目的注释。

语法

`Object.VersionComments [= cComments]`

参数描述

`cComments`

指定一个项目的注释。默认值为空字符串。

说明

`VersionComments` 属性对应于“EXE 版本”对话框中“注释”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionCompany](#) 属性, [VersionCopyright](#) 属性, [VersionDescription](#) 属性, [VersionLanguage](#) 属性, [VersionNumber](#) 属性, [VersionProduct](#) 属性, [VersionTrademarks](#) 属性

VersionCompany 属性

一个项目的公司名信息。

语法

```
Object.VersionCompany[=cCompanyName]
```

参数描述

`cCompanyName`

指定一个项目的公司名信息。默认值为空字符串。

说明

`VersionCompany` 属性对应于“EXE 版本”对话框中“公司名”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments](#) 属性, [VersionCopyright](#) 属性, [VersionDescription](#) 属性, [VersionLanguage](#) 属性, [VersionNumber](#) 属性, [VersionProduct](#) 属性, [VersionTrademarks](#) 属性

VersionCopyright 属性

一个项目的版权信息。

语法

```
Object.VersionCopyright[=cCopyRight]
```

参数描述

`cCopyright`

指定一个项目的版权信息。默认值为空字符串。

说明

`VersionCopyright` 属性对应于“EXE 版本”对话框中“合法版权”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments](#) 属性, [VersionCompany](#) 属性, [VersionDescription](#) 属性, [VersionLanguage](#) 属性, [VersionNumber](#) 属性, [VersionProduct](#) 属性, [VersionTrademarks](#) 属性

`VersionDescription` 属性

一个项目的说明。

语法

`Object.VersionDescription[= cDescription]`

参数描述

`cDescription`

指定一个项目的说明。默认值为空字符串。

说明

`VersionDescription` 属性对应于“EXE 版本”对话框中“文件说明”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments](#) 属性, [VersionCompany](#) 属性, [VersionCopyright](#) 属性, [VersionLanguage](#) 属性, [VersionNumber](#) 属性, [VersionProduct](#) 属性, [VersionTrademarks](#) 属性

VersionLanguage 属性

一个项目的语言信息。

语法

```
Object.VersionLanguage[=nLanguage]
```

参数描述

nLanguage

指定一个项目的语言 ID。FOXPRO.H 包含一组主要和次要语言的 ID 值，这些值可以组合起来创建一个有效的语言 ID。

例如，下列代码将当前项目的 VersionLanguage 属性设置为 UnitedStatesEnglish（0x09 指定 English，0x0400 指定 UnitedStatesEnglish）：
`Application.ActiveProject.VersionLanguage=0x09+0x0400`

说明

VersionLanguage 属性对应于“EXE 版本”对话框中“语言 ID”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments](#) 属性, [VersionCompany](#) 属性, [VersionCopyright](#) 属性, [VersionDescription](#) 属性, [VersionNumber](#) 属性, [VersionProduct](#) 属性, [VersionTrademarks](#) 属性

VersionNumber 属性

一个项目的连编编号。

语法

`Object.VersionNumber[=cBuildNumber]`

参数描述

cBuildNumber

指定一个项目的连编编号。连编编号的标准格式是 "MMMM.mmmm.bbbb"，其中 MMMM 是主连编编号， mmmm 是次连编编号， bbbb 是连编编号。使用句号 (.) 将项目的连编编号各部分分开。每个连编编号可以是零到四个数字。不要在连编编号中使用字母。将 cBuildNumber 指定为 ".."，可以将次连编编号和连编编号重新设置为零。

如果 AutoIncrement 属性设置为“真” (.T.)，并且您从项目连编了一个可执行文件 (.exe) 或动态链接库 (.dll)，将从连编编号最后部分 (".bbbb") 删除前导零。默认值为空字符串。

说明

VersionNumber 属性对应于“EXE 版本”对话框中“版本编号”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments 属性](#), [VersionCompany 属性](#), [VersionCopyright 属性](#), [VersionDescription 属性](#), [VersionLanguage 属性](#), [VersionProduct 属性](#), [VersionTrademarks 属性](#)

VersionProduct 属性

一个项目的产品名信息。

语法

```
Object.VersionProduct[=cProductName]
```

参数描述

cProductName

指定一个项目的产品名信息。默认值为空字符串。

说明

VersionProduct 属性对应于“EXE 版本”对话框中“产品名”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION\(\)](#), [VersionComments 属性](#), [VersionCompany 属性](#), [VersionCopyright 属性](#), [VersionDescription 属性](#), [VersionLanguage 属性](#), [VersionNumber 属性](#), [VersionTrademarks 属性](#)

VersionTrademarks 属性

一个项目的商标信息。

语法

```
Object.VersionTrademarks[= cTradeMarks]
```

参数描述

cTradeMarks

指定一个项目的商标信息。默认值为空字符串。

说明

VersionTrademarks 属性对应于“EXE 版本”对话框中“合法商标”项的值。

应用于

项目对象

请参阅

[AGETFILEVERSION \(\)](#)，[VersionComments](#) 属性，[VersionCompany](#) 属性，[VersionCopyright](#) 属性，[VersionDescription](#) 属性，[VersionLanguage](#) 属性，[VersionNumber](#) 属性，[VersionProduct](#) 属性

_VFP 系统变量

指向当前运行的 VisualFoxPro 应用程序对象。

语法

`_VFP.PropertyName[=eValue]`

-或-

`_VFP.Method`

参数描述

PropertyName

指定应用程序对象的属性。

eValue

指定属性的值。

Method

指定应用程序对象的方法。

说明

通过 `_VFP`，您可以访问对象集合 (Objectscollection)。

属性

ActiveForm	Application	AutoYield
Caption	DefaultFilePath	Forms
FullName	Height	Left
Name	OLERequestPending Timeout	OLEServerBusyRaiseError
OLEServerBusy Timeout	Parent	StartMode
StatusBar	Top	Version
Visible	Width	

方法

DataToClip	DoCmd	Eval
Help	Quit	RequestData
SetVar		

请参阅

Application 对象, CREATEOBJECT()

View 属性

指定表格控件的查看方式。设计时可用，运行时可读写。

语法

```
Grid.View[=nType]
```

参数描述

nType

若没有拆分窗格（即 Partition 属性设置为 0），则 View 属性设置为：

设置	说明
----	----

0	浏览
1	编辑

若窗格已拆分，则 View 属性的设置为：

设置	说明
----	----

0	浏览（左窗格），浏览（右窗格）
1	浏览（左窗格），编辑（右窗格）
2	编辑（左窗格），浏览（右窗格）
3	更改（左窗格），更改（右窗格）

应用于

表格

请参阅

[Panel 属性](#), [PanelLink 属性](#), [Partition 属性](#)

ViewPortHeight 属性

包含一个表单的视口高度。设计和运行时只读。

语法

Object.ViewPortHeight

说明

视口一个表单在容器中显示的矩形区域。表单容器控制了视口的大小。如果更改表单容器的大小，视口也将改变大小。

视口的属性和方法通常用于 ActiveDocuments 中的表单。如果一个 ActiveDocument 显示了一个带滚动条的表单，则当视口的尺寸小于围绕表单上控件的矩形的大小时，就会在该表单上显示滚动条。

视口高度的度量单位由表单的 ScaleMode 属性设置确定——像素（默认的）或 foxels。

应用于

表单对象

请参阅

[ScaleMode 属性](#), [SetViewPort 方法](#), [ViewPortLeft 属性](#), [ViewPortTop 属性](#), [ViewPortWidth 属性](#)

ViewPortLeft 属性

包含在视口中可见的表单的左边坐标。设计和运行时只读。

语法

Object.ViewPortLeft

说明

视口一个表单在容器中显示的矩形区域。表单容器控制了视口的大小。如果更改表单容器的大小，视口也将改变大小。

视口的属性和方法通常用于 ActiveDocuments 中的表单。如果一个 ActiveDocument 显示了一个带滚动条的表单，则当视口的尺寸小于围绕表单上控件的矩形的大小时，就会在该表单上显示滚动条。

使用 SetViewport 方法可以在运行时刻设置 ActiveDocument 中一个表单的左上坐标。视口高度的度量单位由表单的 ScaleMode 属性设置确定——像素（默认的）或 foxels。

应用于

表单对象

请参阅

[ScaleMode 属性](#), [SetViewPort 方法](#), [ViewPortHeight 属性](#), [ViewPortTop 属性](#), [ViewPortWidth 属性](#)

ViewPortTop 属性

包含在视口中可见的表单的上边坐标。设计和运行时只读。

语法

Object.ViewPortTop

说明

视口一个表单在容器中显示的矩形区域。表单容器控制了视口的大小。如果更改表单容器的大小，视口也将改变大小。

视口的属性和方法通常用于 ActiveDocuments 中的表单。如果一个 ActiveDocument 显示了一个带滚动条的表单，则当视口的尺寸小于围绕表单上控件的矩形的大小时，就会在该表单上显示滚动条。

使用 SetViewport 方法可以在运行时刻设置 ActiveDocument 中一个表单的左上坐标。

视口高度的度量单位由表单的 `ScaleMode` 属性设置确定——像素（默认的）或 `foxels`。

应用于

表单对象

请参阅

[ScaleMode 属性](#), [SetViewPort 方法](#), [ViewPortHeight 属性](#), [ViewPortLeft 属性](#), [ViewPortWidth 属性](#)

ViewPortWidth 属性

包含一个表单的视口宽度。设计和运行时只读。

语法

`Object.ViewPortWidth`

说明

视口一个表单在容器中显示的矩形区域。表单容器控制了视口的大小。如果更改表单容器的大小，视口也将改变大小。

视口的属性和方法通常用于 `ActiveDocuments` 中的表单。如果一个 `ActiveDocument` 显示了一个带滚动条的表单，则当视口的尺寸小于围绕表单上控件的矩形的大小时，就会在该表单上显示滚动条。

视口高度的度量单位由表单的 `ScaleMode` 属性设置确定——像素（默认的）或 `foxels`。

应用于

表单对象

请参阅

[ScaleMode 属性](#), [SetViewPort 方法](#), [ViewPortHeight 属性](#), [ViewPortLeft 属性](#), [ViewPortTop 属性](#)

Visible 属性

指定对象是可见还是隐藏。设计和运行时可用。

语法

```
Object.Visible [= IExpr]
```

参数描述

IExpr

Visible 属性的设置有：

设置

说明

“真” (.T.)	在表单设计器中为默认值。对象可见。
“假” (.F.)	在程序代码中为默认值。对象隐藏。

说明

设置_SCREEN 系统变量的 Visible 属性，在 VisualFoxProforWindows 中不起作用。

即使对象隐藏，在代码中也可以访问该属性。

为了在启动时隐藏对象，请在设计时设置 Visible 属性为“假” (.F.)。如果在代码中设置了 Visible 属性，可以隐藏对象，并在运行时，响应一个特定的事件而显示它。

当表单的 Visible 属性设置为“假” (.F.) 时，表单就隐藏起来，而最近活动的表单集、表单或其他对象则变为活动的。当表单的 Visible 属性设置为“真” (.T.) 时，表单变为可见的。将表单的 Visible 属性设置为“真” (.T.) 并不影响表单的 Order 属性设置。在表单的 Visible 属性设置为“真” (.T.) 时，它并不变为活动的，可使用 Show 方法来激活表单并同时使之可见。

注意 如果表单对象的 Visible 属性设置为“假” (.F.)，则不显示该表单，即使表单集对象的 Visible 属性设置为“真” (.T.)。然而，如果表单集对象的 Visible 属性设置为“假” (.F.)，则所有包含在该表单集中的表单对象都不可见。

示例

下面的示例演示了 Visible 属性在用 AddObject 方法加到表单之后，是如何用来显示控

件的。

用 AddObject 方法可向表单中添加一个“线条”控件和三个命令按钮。将这些控件和按钮的 Visible 属性设置为“真”（.T.），以便在表单中显示。Visible 属性也用来在“线条”控件的倾斜方向改变之前隐藏它，并在其倾斜方向改变之后显示它。

```
frmMyForm=CREATEOBJECT('Form')&&创建一个表单  
frmMyForm.Closable=.F.&&废止控件菜单框
```

```
frmMyForm.AddObject('shpLine','Line')&&在表单上添加线条控件  
frmMyForm.AddObject('cmdCmndBtn1','cmdMyCmndBtn1')&&向上命令按钮  
frmMyForm.AddObject('cmdCmndBtn2','cmdMyCmndBtn2')&&向下命令按钮  
frmMyForm.AddObject('cmdCmndBtn3','cmdMyCmndBtn3')&&退出命令按钮
```

```
frmMyForm.shpLine.Visible=.T.&&使线条控件可见  
frmMyForm.shpLine.Top=20&&指定线条控件起始点所在的行  
frmMyForm.shpLine.Left=125&&指定线条控件起始点所在的列
```

```
frmMyForm.cmdCmndBtn1.Visible=.T.&&向上命令按钮可见  
frmMyForm.cmdCmndBtn2.Visible=.T.&&向下命令按钮可见  
frmMyForm.cmdCmndBtn3.Visible=.T.&&退出命令按钮可见
```

```
frmMyForm.SHOW&&显示表单  
READEVENTS&&进入事件循环
```

```
DEFINECLASScmdMyCmndBtn1ASCOMMANDBUTTON&&创建命令按钮  
Caption='Slant\<Up'&&命令按钮的标题  
Left=50&&命令按钮起始点所在的列  
Top=100&&命令按钮起始点所在的行  
Height=25&&命令按钮高度
```

```

PROCEDURE Click
    ThisForm.shpLine.Visible = .F. && 隐藏线条控件
    ThisForm.shpLine.LineSlant = '/' && 向上倾斜
    ThisForm.shpLine.Visible = .T. && 显示线条控件
ENDDEFINE

DEFINE CLASS cmdMyCmndBtn2 AS CommandButton && 创建命令按钮
    Caption = 'Slant\<Down' && 命令按钮的标题
    Left = 200 && 命令按钮起始点所在的列
    Top = 100 && 命令按钮起始点所在的行
    Height = 25 && 命令按钮高度

    PROCEDURE Click
        ThisForm.shpLine.Visible = .F. && 隐藏线条控件
        ThisForm.shpLine.LineSlant = '\' && 向下倾斜
        ThisForm.shpLine.Visible = .T. && 显示线条控件
    ENDDEFINE

DEFINE CLASS cmdMyCmndBtn3 AS CommandButton && 创建命令按钮
    Caption = '\<Quit' && 命令按钮的标题
    Cancel = .T. && 默认取消按钮 (ESC)
    Left = 125 && 命令按钮起始点所在的列
    Top = 150 && 命令按钮起始点所在的行
    Height = 25 && 命令按钮高度

    PROCEDURE Click
        CLEAR EVENTS && 中止事件循环，关闭表单
    ENDDEFINE

```

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，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，页框，项目对象，_SCREEN，形状，微调，文本框，工具栏
请参阅

[Show 方法](#), [ZOrder 方法](#)

VScrollSmallChange 属性

指定当单击一个表单的滚动箭头时的垂直滚动量。设计和运行时可用。

语法

```
Object.VScrollSmallChange[=nExpression]
```

参数描述

nExpression

指定垂直滚动的增量。度量单位由表单的 `ScaleMode` 属性设置确定。当设置为 3——像素时，默认的增量为 10 个像素。

应用于

表单

请参阅

[HScrollSmallChange 属性](#), [ScaleMode 属性](#), [ScrollBars 属性](#)

WAIT 命令

显示信息并暂停 VisualFoxPro 的执行，按某个键或单击鼠标后继续执行。

语法

WAIT

[cMessageText]

[TO VarName]

[WINDOW [AT nRow, nColumn]]

[NOWAIT]

[CLEAR|NOCLEAR]

[TIMEOUT nSeconds]

参数描述

cMessageText

指定要显示的自定义信息。若省略 *cMessageText* 参数，则 VisualFoxPro 显示默认的信息。如果 *cMessageText* 参数为空字符串('')，则不显示信息，直到按某个键，VisualFoxPro 才结束等待，继续执行程序。

TO VarName

将按下的键保存到变量或数组元素中。如果用 *VarName* 参数指定的变量或数组元素不存在，则创建一个。若按 ENTER、不可打印的键或组合键、或者单击鼠标，那么在 *VarName* 参数中就存储空字符串。

WINDOW

在 VisualFoxPro 主窗口右上角的系统信息窗口中，显示信息。按 CTRL 键或 SHIFT 键可以暂时隐藏该窗口。

ATnRow,nColumn

在 VisualFoxPro 中，指定信息窗口在屏幕上的位置。

NOWAIT

在显示信息后，立即继续执行程序。程序并不等待信息从 VisualFoxPro 主窗口中删除，而是从包含 WAITNOWAIT 的程序行之之后紧接的一行开始继续执行。如果省略了 NOWAIT 参数，程序直到信息从 VisualFoxPro 主窗口中删除，才开始执行。删除信息可按某一键或单击鼠标。

CLEAR

在程序中，从 VisualFoxPro 主窗口中删除 VisualFoxPro 的系统窗口或 WAIT 信息窗口。例如，如果发出 SETTALKWINDOW 命令，则索引、排序等等过程中的对话都指向 VisualFoxPro 系统窗口，若按某一键或单击鼠标，可以删除该窗口。发布 WAITCLEAR 命令可以在程序中删除窗口。

NOCLEAR

指定 WAIT 信息窗口保留在 VisualFoxPro 主窗口中，直到发布 WAITCLEAR

或另一条命令 WAITWINDOW，或者直到显示 VisualFoxPro 系统信息。

TIMEOUTnSeconds

指定在中断 WAIT 命令之前，等待键盘或鼠标输入的秒数，*nSeconds* 参数指定经过的秒数（允许有分秒 (fractionalseconds)）。如果 TIMEOUT 不是 WAIT 命令的最后一个子句，则 VisualFoxPro 产生语法错误信息。

说明

如果在 Windows 版本的 VisualFoxPro 中出现了 WAIT 信息，按下 SHIFT 或 CTRL 键就可以隐藏包括 WAIT 信息在内的所有窗口。

请参阅

ACCEPT

WBORDER () 函数

确定活动窗口或指定窗口是否有边框。

语法

WBORDER([WindowName])

返回值类型

逻辑型

参数描述

WindowName

指定需要 WBORDER() 返回其信息的窗口名。

如果省略窗口名，WBORDER() 返回一个逻辑值，指明活动输出窗口是否有边框。

说明

如果指定窗口有边框，WBORDER() 返回“真” (.T.)，否则返回“假” (.F.)。默认情况下，窗口有边框。在 DEFINEWINDOW 中包括 NONE 子句，可以创建没有边框的窗口。

请参阅

[ACTIVATEWINDOW, DEFINEWINDOW](#)

WCHILD () 函数

返回父窗口中子窗口的数目，或者按照子窗口在父窗口中排放的顺序返回子窗口的名称。

语法

WCHILD([WindowName][nChildWindow])

返回值类型

字符型或数值型

参数描述

WindowName

指定一个非活动输出窗口，WCHILD()函数返回它的子窗口数目。如果只包含窗口名而不包含 *nChildWindow*，则返回指定窗口中子窗口的数目。

如果既包含了 *WindowName*，又包含了 *nChildWindow*，则 WCHILD()函数返回指定窗口中子窗口的名称。这时需用逗号将两个参数分隔开。

也可以在 *WindowName* 中包括空字符串，指定 VisualFoxPro 的主窗口。

nChildWindow

指定一个数值表达式，在省略 *WindowName* 参数时，返回活动输出窗口中的子窗口名称。

数值表达式 *nChildWindow* 可以是 0 或任一正值。如果 *nChildWindow* 为 0，则活动输出窗口中排放在最底部的子窗口名称被返回。

若 *nChildWindow* 为某一正数，WCHILD()函数返回子窗口序列中下一个子窗口的名称。如果重新发出带有正数的 WCHILD()函数，就返回子窗口序列中再下一个子窗口名称。如果调用 WCHILD()函数的次数多于父窗口中子窗口的数目，则返回空字符串。有关窗口序列的更多内容，请参阅 ACTIVATEWINDOW。

注意 若激活 FoxPro 主窗口，则所有窗口都成为主窗口的子窗口。在 VisualFoxPro 中，不在 VisualFoxPro 主窗口的边界上停放的工具栏也是主窗口的子窗口。发出带有正数参数的一系列 WCHILD()函数，就会返回活动窗口和工具栏的名字。

如果既包含 *WindowName*，又包含了 *nChildWindow*，这时需用逗号将两个参数分隔开

来。

说明

可以创建一个窗口（父窗口），并将其他窗口（子窗口）放置在其中。在 `DEFINERWINDOW` 中包含 `IN` 或 `INWINDOW` 子句，可在父窗口中创建子窗口。在父窗口中创建并激活的子窗口不能移出父窗口之外。如果父窗口移动，子窗口也随之移动。

如果 `WCHILD()` 函数不带任何参数，则返回活动输出窗口中子窗口的数目。

请参阅

[ACTIVATEWINDOW](#), [DEFINERWINDOW](#), [WPARENT\(\)](#)

WCOLS () 函数

返回活动窗口或指定窗口中的列数。

语法

`WCOLS([WindowName])`

返回值类型

数值型

参数描述

WindowName

指定要 WCOLS()函数返回其列数的窗口。在 VisualFoxPro 中，可以指定工具栏的名字。如果不指定窗口，则返回活动输出窗口的列数。如果没有活动窗口，WCOLS()返回 VisualFoxPro 主窗口的列数。

如果系统窗口激活并且可见或隐藏，则可以在 WCOLS()函数中指定系统窗口名（例如命令窗口(Command)、查看窗口(View)、浏览窗口(Browse)等）。在 VisualFoxPro 中，如果工具栏激活并可见或隐藏，则可以在 WCOLS()函数中指定工具栏名称（例如常用工具栏、调色板工具栏等）。如果指定尚未激活的系统窗口或工具栏的名称，VisualFoxPro 产生错误信息。但调试窗口(Debug)例外，只要打开调试窗口，不论它可见、隐藏或关闭，都可以在 WCOLS()中包含其名称。

也可以包含空字符串作为窗口名称，从而返回 VisualFoxPro 主窗口的列数。

可以在返回窗口位置或大小的函数 WLCOL(), WLROW(), 和 WROWS()中，使用空字符串指定 VisualFoxPro 主窗口。

说明

在 VisualFoxPro 中，WCOLS()函数的返回值取决于指定窗口的字体，许多字体可以用一系列大小显示，有些能够按比例保持间距。列对应于当前字体中字母的平均宽度。有关详细内容，请参阅“字体概述”。

示例

下面的示例将一个短的输出信息在未知大小窗口的最后一列中居中对齐。

```
CLEAR  
DO SendMesg WITH 'Message', WCOLS(), WROWS()-1
```

SendMsg

```
PROCEDURE SendMsg  
PARAMETERS gcMsg, gnCol, gnRow  
STORE (gnCol-LEN(gcMsg))/2 TO gnCol  
@gnRow, gnCol SAY gcMsg  
RETURN
```

请参阅

[SCOLS\(\)](#), [SROWS\(\)](#), [WLCOL\(\)](#), [WLROW\(\)](#), [WROWS\(\)](#)

WEEK () 函数

返回值类型 从日期或日期时间表达式返回一个表示一年中的星期的数。

语法

WEEK (dExpression|tExpression[,nFirstWeek][,nFirstDayOfWeek])

返回值类型

数值型

参数描述

dExpression|tExpression

为 WEEK()返回的一年中的星期指定一个日期或日期时间表达式。

如果省略了可选的 nFirstWeek 与 nFirstDayOfWeek 参数描述,WEEK()将用 Sunday 作为一个星期的开始。

nFirstWeek

指定一年的第一个星期的要求。nFirstWeek 可以是下列值。

nFirst Week	说明
------------------------	-----------

0	WEEK()返回在“选项”对话框“国际”选项卡“年的第一星期”列表框中当前选择的任意一周。
1	第一周包含一月一日。当省略 <i>nFirstweek</i> 时，这是默认值。
2	第一周的大半（四天）在本年中。
3	第一周有七天。

nFirstDayOfWeek

指定一周的第一天。nFirstDayOfWeek 可以是下列值：

nFirstDayOf Week	Description
-----------------------------	--------------------

0	WEEK()返回在“选项”对话框“国际”选项卡“启始周”列表框中当前选择的任意一天。
---	--

续表

1	星期天。当省略 nFirstDayOfWeek 时，这是默认值，并且在早期版本的 FoxPro 中使用的一周的第一天。
2	星期一
3	星期二
4	星期三
5	星期四
6	星期五
7	星期六

说明

WEEK() 返回 1-53 之间的一个值，该值代表了一年的周。例如，对于一年的第一周，WEEK() 返回 1，对于一年的第二周，WEEK() 返回 2 等等。注意一周可以跨年，即一年的第一周可以在本年，也可以在上一年。

示例

下例显示了此时与一个指定日期处于一年中的第几个星期。

```
CLEAR  
?WEEK (DATE())  
?WEEK ({^1998-02-16})&&显示 8
```

请参阅

[CDOW\(\)](#), [DAY\(\)](#), [DOW\(\)](#), [SETFDOW](#), [SETFWEEK](#), [SYS\(\)](#) 函数概述

WEXIST () 函数

确定所指定的用户自定义窗口是否存在。

语法

WEXIST(WindowName)

返回值类型

逻辑型

参数描述

WindowName

指定用户自定义窗口的名称。

也可以指定 VisualFoxPro 系统窗口的名称（如命令窗口(Command)、查看窗口(View)、浏览窗口(Browse)等）；在 VisualFoxPro 中，还可以指定工具栏的名称。如果所指定的系统窗口或工具栏可见或隐藏，则 WEXIST()函数返回“真”(.T.)。如果所指定的系统窗口或工具栏已关闭，则返回“假”(.F.)。

命令窗口(Command)和调试窗口(Debug)是两个例外。在 WEXIST()中包含命令窗口名时，总是返回“真”(.T.)。如果曾经打开过调试(Debug)窗口，即使已经关闭，WEXIST()函数仍返回“真”(.T.)。

说明

如果所指定的用户自定义窗口是用 DEFINEWINDOW 命令创建的，则 WEXIST()函数

返回“真”(.T.)，窗口不一定活动或可见，但必须存在。

示例

```
DEFINERWINDOWwScreen1FROM10,10TO20,69
DEFINERWINDOWwScreen2FROM1,0TO19,79
CLEAR

?WEXIST('wScreen1')&&显示.T.
STORE'wScreen2'TOgcWinName
?WEXIST('win_name')&&显示.F.
?WEXIST(gcWinName)&&显示.T.
RELEASEWINDOWSwScreen1,wScreen2
```

请参阅

[ACTIVATEWINDOW](#), [DEFINERWINDOW](#), [WONTOP\(\)](#), [WOUTPUT\(\)](#), [WVISIBLE\(\)](#)

WFONT () 函数

返回 VisualFoxPro 窗口当前字体的名称、大小或字形。

语法

```
WFONT(nFontAttribute [,WindowName])
```

返回值类型

字符型和数值型

参数描述

nFontAttribute

指定要返回的字体属性。

如果 *nFontAttribute* 是 1，WFONT()函数返回活动窗口或指定窗口的当前字体名。在 FoxProforMS-DOS 中，WFONT()函数总是返回“FoxFont”。

如果 *nFontAttribute* 是 2，WFONT()函数返回字体大小。

如果 *nFontAttribute* 是 3，WFONT()返回标识字形的代码。

字形代码为代表当前字形的单个字符或一组字符。例如，若 WFONT(3)返回 BI，则当前字形就是粗斜体。

下表列出了每种字形的代码：

字符

字形

B	粗体
I	斜体
N	常规
O	轮廓
Q	不透明
S	阴影
-	删除线
T	透明
U	下划线

WindowName

指定要确定当前字体大小或字形的窗口名称。在 VisualFoxPro 中，也可以指定工具栏的名称。若包含的是空字符串，则返回 VisualFoxPro 主窗口的当前字体的名称、大小或字形。

WindowName 可以是 DEFINEWINDOW 命令创建的用户自定义窗口名，或者文本编辑与备注编辑窗口名。

也可以包含系统窗口的名称（如查看(View)、跟踪(Trace)、调试(Debug)等窗口）。

WFONT()函数只能为打开的且当前可见或隐藏的系统窗口返回字体属性。如果指定的系统窗口关闭，则 VisualFoxPro 显示错误信息。

如果省略参数 *WindowName*，WFONT()函数返回活动输出窗口的当前字体、字体大小或字形。

示例

下面的示例创建了名为 wFontChar 的用户自定义窗口。窗口被激活并且显示它的字体特征。然后为主 VisualFoxPro 窗口显示字体特征。

```
CLEAR
DEFINEWINDOWwFontChar;
  FROM1,1TO3,35;
  FONT'MSSANSSERIF',8;
  STYLE'BI'          &&用字体和样式定义窗口
ACTIVATEWINDOWwFontChar
?WFONT(1),WFONT(2),WFONT(3)  &&wFontChar窗口
ACTIVATESCREEN
?
```

```
?  
?  
?'FontcharacteristicsforthefontwFontChar'  
?  
?  
?WFONT(1,''),WFONT(2,''),WFONT(3,'') &&VisualFoxPro 主窗口  
?  
?'FontcharacteristicsforthefontwFontChar'  
WAITWINDOW  
RELEASEWINDOWwFontChar  
CLEAR
```

请参阅

[FONTMETRIC\(\)](#), [GETFONT\(\)](#), [SYSMETRIC\(\)](#), [TXTWIDTH\(\)](#)

WhatsThisButton 属性

指定在表单的标题栏中是否显示“这是什么”按钮。运行时只读。

语法

Form.WhatsThisButton[=lExpr]

参数描述

IExpr

取下列一个值：

设置

说明

“真”

打开对“这是什么”按钮的显示。

(.T.)

“假”

(默认值) 关闭对“这是什么”按钮的显示。

(.F.)

说明

如果 WhatsThisButton 属性设置为“真”(.T.)，并且下列情况为真，则在表单的标题栏中不显示“这是什么”按钮：

- 表单的 WhatsThisHelp 属性设置为“假”(.F.)。
- 表单的 BorderStyle 属性设置为 0 (无)。
- 表单的 MinButton 属性设置为“真”(.T.)。
- 表单的 MaxButton 属性设置为“真”(.T.)。

应用于

表单

请参阅

[BorderStyle](#) 属性, [MaxButton](#) 属性, [MinButton](#) 属性, [ShowWhatsThis](#) 方法, [TitleBar](#) 属性, [WhatsThisMode](#) 方法, [WhatsThisHelp](#) 属性, [WhatsThisHelpID](#) 属

性

WhatsThisHelp 属性

指定上下文相关帮助使用“这是什么”帮助还是使用 SETHELP 指定的 Windows 帮助文件。

语法

```
Form.WhatsThisHelp[=IExpr]
```

参数描述

IExpr

取下列一个值：

设置

说明

“真”

(.T.)

表单使用三种“这是什么”帮助技术之一打开由 SETHELP 指定的 Windows 帮助文件，并且显示由 WhatsThisHelpID 属性指定的主题。

“假”

(.F.)

(默认值) 表单使用 F1 键由 SETHELP 指定的 Windows 帮助文件。显示由 HelpContextID 属性指定的主题。

说明

三种“这是什么”帮助技术是：

1. 在表单的标题栏显示一个“这是什么”按钮。当单击这个按钮时，鼠标指针变成带有问号的箭头。当单击一个控件时，所显示的帮助主题由该控件的 WhatsThisHelpID 属性指定。
2. 激活 WhatsThisMode 方法，鼠标指针变成带有问号的箭头。当单击一个控件时，所显示的帮助主题由该控件的 WhatsThisHelpID 属性指定。
3. 激活一个控件的 ShowWhatsThis 方法。所显示的帮助主题由该控件的 WhatsThisHelpID 属性指定。

注意在 VisualFoxPro6.0 中，如果将帮助设置为一个 HTML 帮助文件（例如，Foxhelp.chm），则在 HTML 帮助查看器中显示帮助主题。如果将帮助设置为一个 Windows 帮助文件（例如，Foxhelp.hlp），则在控件旁边的一个小弹出窗口中显示帮助主题。

应用于

表单

请参阅

[HelpContextID 属性](#), [SETHelp](#), [ShowWhatsThis 方法](#), [WhatsThisButton 属性](#), [WhatsThisMode 方法](#), [WhatsThisHelpID 属性](#)

WhatsThisHelpID 属性

指定一个帮助主题上下文 ID，以便为一个对象提供“这是什么”帮助。设计和运行时可用。

语法

Object.WhatsThisHelpID[=`nContextID`]

设置

`nContextID`

指定帮助文件中一个主题的上下文 ID。`nContextID` 可以是负数、零或正数。

下表列出了 `nContextID` 的值，以及当使用问号鼠标指针单击对象时所显示的内容：

nContextID	说明
负数	（默认值，-1）弹出显示“该对象没有相关的帮助主题”消息的“这是什么”帮助。
零	VisualFoxPro 向上搜索对象的层次，找到第一个 WhatsThisHelpID 属性值为正数的对象。如果找到了，则显示这个“这是什么”帮助主题。如果没有找到，则不显示任何“这是什么”帮助主题。

续表

正数 显示对应于 nContextID 指定的上下文 ID 的“这是什么”帮助主题。如果没有帮助主题具有 nContextID 指定的上下文 ID，则弹出显示“该对象没有相关的帮助主题”消息的“这是什么”帮助。

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，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，形状，微调，文本框，计时器，工具栏

请参阅

[HelpContextID 属性](#), [SETHelp](#), [ShowWhatsThis 方法](#), [WhatsThisButton 属性](#), [WhatsThisHelp 属性](#), [WhatsThisMode 方法](#)

WhatsThisMode 方法

显示“这是什么”帮助的问号鼠标指针，并且启用“这是什么”帮助模式。

语法

[Form.]Object.WhatsThisMode

说明

调用 `WhatsThisMode` 方法，会将鼠标指针变成带有问号的箭头，并且启用“这是什么”帮助模式。单击一个对象，会显示该对象的 `WhatsThisHelpID` 属性指定的“这是什么”帮助主题。按 `ESC` 键会废止“这是什么”帮助模式，并且将鼠标指针恢复原样。

应用于

表单

请参阅

[HelpContextID 属性](#), [SETHelp](#), [ShowWhatsThis 方法](#), [WhatsThisButton 属性](#), [WhatsThisHelp 属性](#), [WhatsThisHelpID 属性](#)

When 事件

在控件接收焦点之前此事件发生。

语法

```
PROCEDURE Control.When  
[LPARAMETERSnIndex]
```

参数描述

nIndex

唯一地标识控件数组中的某个控件。

说明

如果 When 事件返回“真”(.T.)，默认控件接收到焦点；如果返回“假”(.F.)，控件未接收到焦点。在控件获得焦点时，事件顺序为：

1. When 事件

2. GotFocus 事件

对于列表框控件，每当用户单击列表中的项或用箭头键移动使焦点在项之间移动时，When 事件发生。

对所有其他控件，当试图把焦点移动到控件上时，When 事件发生。

应用于

复选框，组合框，命令按钮，命令组，编辑框，表格，列表框，选项按钮，选项组，微调，文本框

请参阅

[GotFocus 事件](#)，[Valid 事件](#)

Width 属性

指定对象的宽度。设计和运行时可用。

语法

```
[Object.]Width[=nWidth]
```

参数说明

nWidth

指定对象的宽度。

说明

对对象的整个区域进行操作或计算（例如，调整对象大小或移动对象）时，可使用 Height，Width，Left 和 Top 属性。

对表单对象和控件，此属性的值随用户或代码所定义的大小而改变。所有对象的 Width 属性的最大值取决于系统。

对表单对象，Width 属性指定表单对象的外部宽度，但不包括边框。

对于控件，Width 属性从控件边框的中心计算，以使不同边框的控件能够正确地对齐。

注意当应用于列对象中包含的控件时，Width 属性只读。Width 属性的度量单位由 ScaleMode 属性的设置值决定。

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，自定义，编辑框，表单，表格，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，页框，_SCREEN，形状，微调，文本框，计时器，工具栏
请参阅

[Height 属性](#),[Left 属性](#),[Move 方法](#),[ScaleMode 属性](#),[Top 属性](#)

WindowList 属性

包含此属性是为了提供与 READ 的向后兼容性。用“表单设计器”代替 READ。

WindowState 属性

指定表单窗口在运行时是否可以最大化或最小化。设计和运行时可用。

语法

```
[Object.]WindowState[=nState]
```

参数说明

nState

WindowState 属性的设置有：

设置	说明
0	正常
1	最小化（最小为一个图标）。如果当退出时，VisualFoxPro 主窗口最小化了，则在退出之前不显示 VisualFoxPro 主窗口。如果您的应用程序在退出之前显示一个对话框，请在显示这个对话框之前将 <code>_SCREEN.WindowState</code> 设置为 0。
2	最大化（扩大到填满屏幕）

应用于

表单，`_SCREEN`

请参阅

[Height 属性](#) , [Left 属性](#) , [Top 属性](#) , [Width 属性](#)

`_WINDOWS` 系统变量

如果正在使用 VisualFoxPro，则包含“真”(.T.)。

语法

`_WINDOWS=!Expression`

说明

如果正在使用的是 VisualFoxPro，_WINDOWS 系统变量包含“真”(.T.)；如果正在使用的是 FoxPro 的其他平台版本，则 _WINDOWS 变量包含“假”(.F.)。

_WINDOWS 所包含的值不能用 STORE 命令或赋值号 (=) 来更改。

请参阅

[_DOS](#), [_MAC](#), [_UNIX](#), [VERSION\(\)](#)

WindowType 属性

在执行 DOFORM 命令时，指定表单集或表单对象的动作。设计和运行时可用。

语法

`Object.WindowType[=nType]`

参数说明

nType

对于表单集对象，WindowType 属性的设置有：

设置

说明

-
- | | |
|---|---|
| 0 | 无模式。 |
| 1 | 模式。其他表单对象不能成为活动的，菜单也为非活动的。表单集中所有的表单对象都活动。 |
| 2 | 读。表单集的行为象是被 READ 命令激活一样。程序执行到 Show 方法或 DOFORM 命令时停止。当表单不活动时，程序继续执行（包含此属性是为了提供向后兼容性）。仅用于从当前版本的 FoxPro 中转换的表单。 |
| 3 | 读模式。表单集的行为象是被 READ 命令的 MODAL 子句激活一样，程序执行到 Show 方法或 DOFORM 命令时停止。在 WindowList 属性中指定的任何表单对象都可用，但是其他表单对象和菜单不可用（包含此属性是为了提供向后兼容性。仅用于从当前版本的 FoxPro 中转换的表单。） |

对于表单对象， WindowType 属性的设置有：

设置

说明

-
- | | |
|---|--|
| 0 | 无模式。 |
| 1 | 模式。其他表单对象不能成为活动的，菜单也不活动。而表单集中的所有表单对象则是活动的。 |

说明

一旦显示了窗口，则不能更改 WindowType 设置。

Show 方法带有能够忽略 WindowType 设置的参数。

注意表单集的 WindowType 设置忽略它所包含的表单对象各自的 WindowType 设置。例如，若表单集的 WindowType 属性设置为 0，则它包含的所有表单对象则都成为无模式的，不管单个表单的 WindowType 属性的设置情况。

应用于

表单，表单集，_SCREEN

请参阅

DOFORM, READEVENTS, Show 方法, WindowList 属性

WITH...ENDWITH 命令

指定对象的多个属性。

语法

WITHObjectName

[.cStatements]

ENDWITH

参数描述

ObjectName

指定对象名。 *ObjectName* 可以是对象名或者对象引用。

.cStatements

该参数可以由大量的 VisualFoxPro 命令组成，这些命令用来指定 *ObjectName* 的属性。在 *cStatement* 之前加一个点号，表明它为 *ObjectName* 的属性。

说明

WITH...ENDWITH 提供了为单个对象指定多个属性的方便途径。

注意，也可以在 WITH...ENDWITH 结构中执行方法。

示例

下面的示例创建了一个自定义的类名 Employee。用 CREATEOBJECT() 创建了类

Employee 后，WITH...ENDWITH 可用来为该类设置多个属性。然后显示属性值。

```
moemployee=CREATEOBJECT('employee')
```

```
WITHmoemployee  
  .First_Name='John'  
  .Last_Name='Smith'  
  .Address='16MapleLane'  
  .HireDate={^1998-02-16}  
ENDWITH
```

```
CLEAR  
?moemployee.First_Name+ "  
??moemployee.Last_Name  
?moemployee.Address  
?moemployee.HireDate
```

```
DEFINECLASSemployeeASCUSTOM
    First_Name=SPACE(20)
    Last_Name=SPACE(20)
    Address=SPACE(30)
    HireDate={--}
ENDDEFINE
```

请参阅

[::ScopeResolutionOperator](#), [ADDCLASS](#), [CREATECLASS](#), [CREATECLASSLIB](#), [CREATEOBJECT\(\)](#), [GETOBJECT\(\)](#), [MODIFYCLASS](#), [RELEASECLASSLIB](#), [SETCLASSLIB](#)

_WIZARD 系统变量

包含 VisualFoxPro 向导应用程序的名称。

语法

```
_WIZARD=cProgramName
```

参数描述

cProgramName

指定向导应用程序。如果向导应用程序不在当前 VisualFoxPro 默认目录下，

可在应用程序名中包括路径。

也可以使用以下语法，在 VisualFoxPro 配置文件中指定向导应用程序：

```
_WIZARD=cProgramName
```

说明

当选定了要运行的向导时，VisualFoxPro 将向导的名称存放在 _WIZARD 系统变量之中。在默认情况下，_WIZARD 包含 WIZARD.APP，安装在 VisualFoxPro 目录下，可以为向导应用程序指定不同的名字。

请参阅

[系统变量概述](#)

WLAST () 函数

返回在当前窗口之前活动的窗口名称，或者确定指定窗口在当前窗口之前是否活动。

语法

```
WLAST([WindowName])
```

返回值类型

字符型或逻辑型

参数描述

WindowName

指定 WLAST()要处理的窗口。在 VisualFoxPro 中，也可以指定工具栏名。如果指定的窗口在当前窗口之前活动，则 WLAST()返回“真”(.T.)，否则返回“假”(.F.)。如果所指定的窗口不存在，也返回“假”(.F.)。

若省略 WindowName，则返回在当前窗口之前活动的窗口的名称。

说明

如果在当前窗口之前活动的窗口为调试窗口、跟踪窗口或命令窗口，则 WLAST()返回空字符串。当调试一个使用 WLAST()函数的程序时，使这些窗口活动并不影响程序的执行。

请参阅

[ACTIVATEWINDOW](#),[DEFINEWINDOW](#),[WEXIST\(\)](#),[WONTOP\(\)](#),[WOUTPUT\(\)](#),[WVISIBLE\(\)](#)

WLCOL () 函数

返回活动窗口或指定窗口左上角的列坐标。

语法

WLCOL([WindowName])

返回值类型

数值型

参数描述

WindowName

指定由 WLCOL() 返回列坐标的窗口。在 VisualFoxPro 中，也可以指定工具栏名。如果省略 *WindowName*，WLCOL() 返回活动输出窗口的列坐标。也可以令 *WindowName* 参数为空字符串，以指定 VisualFoxPro 主窗口。

在 VisualFoxPro 中，如果没有活动窗口，WLCOL() 返回相对于 Windows 桌面或 Macintosh 桌面的 FoxPro 主窗口的列坐标。

如果打开了系统窗口（如 Command（命令窗口）、View（查看窗口）、Browse（浏览窗口）等），并且该窗口是可见的或隐藏的（其名称出现在“窗口”菜单内），则可以在 WLCOL() 中包括它的名称。如果所指定的系统窗口已经关闭，FoxPro 会产生错误信息，但调试窗口(Debug)例外。如果打开了调试窗口，不论是否可见或关闭，都可以在 WLCOL() 中包含它的名称。

说明

由 WLCOL() 返回的值取决于当前视频显示方式，且相对于 VisualFoxPro 主窗口。显示方式可以用 SETDISPLAY 来更改。

窗口可以放置在 VisualFoxPro 主窗口以外。如果指定窗口在 VisualFoxPro 主窗口左侧，WLCOL() 返回负值。指定窗口的左边框在 VisualFoxPro 主窗口右侧时，则返回大于 VisualFoxPro 主窗口宽度的正值。

请参阅

SETDISPLAY, WCOLS(), WLROW(), WROWS()



返回总目录

WLROW() 函数

WMAXIMUM() 函数

WMINIMUM() 函数

WONTOP() 函数

WordWrap 属性

WOUTPUT() 函数

WPARENT() 函数

_WRAP 系统变量

WREAD() 函数

WriteExpression 方法

WriteMethod 方法

WROWS() 函数

WTITLE() 函数

WVISIBLE() 函数

YEAR() 函数

ZAP 命令

ZoomBox 属性

ZOOM WINDOW 命令

ZOrder 方法

WLROW () 函数

返回活动窗口或指定窗口左上角的行坐标。

语法

WLROW([WindowName])

返回值类型

数值型

参数描述

WindowName

指定由 WLROW() 返回行坐标的窗口，在 Visual FoxPro 中，也可以指定工具栏名。如果省略 *WindowName*，WLROW() 返回活动输出窗口的行坐标。也可以令 *WindowName* 为空字符串，以指定 Visual FoxPro 主窗口。

在 Visual FoxPro 中，如果没有活动窗口，WLROW() 返回相对于 Windows 桌面的 Visual FoxPro 主窗口的行坐标。

如果打开了系统窗口（如命令窗口、查看窗口、浏览窗口等），并且该窗口可见或隐藏（名称出现在“窗口”菜单里），则可以在 WLROW() 中包含它的名称。如果所指定的系统窗口已经关闭，FoxPro 会产生错误，但调试窗口例外，如果已打开此窗口，

不论是否可见或者关闭，都可以在 WLROW() 中包含它的名称。

说明

由 WLROW() 返回的值取决于当前视频显示方式，且行坐标相对于 Visual FoxPro 主窗口。显示模式可以由 SET DISPLAY 更改。

窗口可以位于 Visual FoxPro 的主窗口以外。如果窗口顶部在 Visual FoxPro 主窗口以上，WLROW() 返回负值。如果窗口的顶部在主窗口以下，则返回大于 Visual FoxPro 主窗口高度的正值。

请参阅

[SET DISPLAY](#), [WCOLS\(\)](#), [WLCOL\(\)](#), [WROWS\(\)](#)

WMAXIMUM () 函数

确定活动窗口或指定窗口是否最大化。

语法

WMAXIMUM([WindowName])

参数描述

WindowName

指定 WMAXIMUM() 要处理的窗口名。可以指定 Visual FoxPro 系统窗口名（如 Command（命令窗口）、View（查看窗口）、Browse（浏览窗口）等）。

如果省略 *WindowName*，WMAXIMUM() 函数返回活动窗口是否最大化的逻辑值。可以令 *WindowName* 为空字符串，来指定 Visual FoxPro 主窗口。

说明

窗口可以扩大到充满 Visual FoxPro 主窗口。在 Visual FoxPro 中，默认的容器窗口是主窗口。

用 DEFINE WINDOW 创建的用户自定义窗口可以最大化，但须在其定义中包括 ZOOM 关键字。

如果当前窗口或指定窗口已最大化，则 WMAXIMUM() 返回“真”(.T.)，否则返回“假”(.F.)。在 Visual FoxPro 中，也可以包括工具栏名，由于工具栏不能最大化，所以包括工具栏名时 WMAXIMUM() 总是返回“假”(.F.)。

请参阅

[DEFINE WINDOW](#), [WMINIMUM\(\)](#), [ZOOM WINDOW](#)

WMINIMUM () 函数

确定活动窗口或指定窗口是否最小化。

语法

WMINIMUM([WindowName])

返回值类型

逻辑型

参数描述

WindowName

指定 WMINIMUM() 要处理的窗口名。可以指定 Visual FoxPro 系统窗口名（如 Command（命令窗口）、View（查看窗口）、Browse（浏览窗口）等）。

如果省略 *WindowName*，WMINIMUM() 返回指示活动窗口状态的逻辑值。也可以令 *WindowName* 为空字符串，来指定 Visual FoxPro 主窗口。

说明

在 Visual FoxPro 中，窗口可以缩为一个图标。

对于 DEFINE WINDOW 创建的用户自定义窗口，只有在窗口定义中包含了

MINIMIZE 关键字时才可以最小化。

如果当前窗口或指定窗口已最小化，则 WMAXIMUM() 返回“真”(.T.)，否则返回“假”(.F.)。在 Visual FoxPro 中，也可以包括工具栏名，由于工具栏不能最小化，所以包括工具栏名时 WMAXIMUM() 总是返回“假”(.F.)。

请参阅

DEFINE WINDOW, WMAXIMUM(), ZOOM WINDOW

WONTOP () 函数

确定活动窗口或指定窗口是否在所有其他窗口之前。

语法

WONTOP([WindowName])

返回值类型

字符型或逻辑型

参数描述

WindowName

指定 WONTOP() 要处理的窗口的名称。可以指定用 DEFINE WINDOW 创

建的用户自定义窗口或系统窗口（如 Command（命令窗口）、Data Session（数据集窗口）、Browse（浏览窗口）等）。如果省略 *WindowName*，WONTOP() 返回最前面的窗口名。也可以令 *WindowName* 为空字符串，来指定 Visual FoxPro 主窗口。

如果调试、跟踪或命令窗口在最前面，而且没有指定窗口名，则 WONTOP() 返回空字符串。当调试一个使用了 WONTOP() 函数时，将这些窗口放置前面并不影响程序的执行。

说明

如果指定的窗口在最前面，WONTOP() 返回“真”(.T.)，如果不在最前面或者不存在，则返回“假”(.F.)。

示例

下面的示例使用 WONTOP() 来显示最前面的窗口名。在运行此程序之前，先打开一些窗口（浏览窗口、查看窗口等），然后再运行此例程。

```
IF NOT EMPTY(WONTOP())
    WAIT WINDOW 'Frontmost window: ' + WONTOP()
ELSE
    WAIT WINDOW 'No windows are open'
ENDIF
```

请参阅

[ACTIVATE WINDOW](#), [DEFINE WINDOW](#), [WEXIST\(\)](#), [WOUTPUT\(\)](#), [WVISIBLE\(\)](#)

WordWrap 属性

在调整 AutoSize 属性为“真”(T.) 的标签控件大小时，指定是否在垂直方向或水平方向放大该控件，以容纳 Caption 属性指定的文本。设计和运行时可用。

语法

```
[Form.]Label.WordWrap[ = IExpr]
```

参数描述

IExpr

WordWrap 属性设置为：

设置

说明

“真” (.T.)	文本自动换行，标签在垂直方向缩放到恰好容纳文本和字体大小，而水平方向的尺寸不更改。
“假” (.F.)	(默认值) 文本不自动换行，标签在水平方向上缩放到恰好容纳文本长度，且在竖直方向缩放到恰好容纳字体大小和行数。竖直方向的尺寸不更改。

说明

可以用这个属性来确定标签如何显示其内容。例如，动态变化的图像或许有标签，该

标签同时包含内容可变的文本，为了保持标签在水平方向上大小不变，并同时允许增减文本，则须设置 WordWrap 属性为“真”(.T.)。

如果要想让标签只在水平方向扩大，可设置 WordWrap 为“假”(.F.)。如果不想更改标签大小，可将 AutoSize 设置为“假”(.F.)。

注意 如果设置 WordWrap 属性为“真”(.T.)，则忽略 AutoSize 属性。

应用于

标签

请参阅

[AutoSize 属性](#), [Caption 属性](#)

WOUTPUT () 函数

确定输出是否定向到活动窗口或指定窗口。

语法

WOUTPUT([WindowName])

返回值类型

逻辑型和字符型

参数描述

WindowName

指定 WOUTPUT() 要处理的输出窗口。输出不能定向到系统窗口或工具栏中。如果省略 *WindowName*，WOUTPUT() 返回当前输出窗口名；也可以在 *WindowName* 中包含空字符串，指定 Visual FoxPro 主窗口为输出窗口。

WOUTPUT() 在输出定向到 Visual FoxPro 主窗口时返回空字符串。

说明

如果所指定的用户自定义窗口为活动输出窗口，则 WOUTPUT() 返回“真”(.T.)，如果所指定的窗口未定义或者是系统窗口，则该函数返回“假”(.F.)。用 ACTIVATE WINDOW 命令激活的最近的用户自定义窗口为活动输出窗口。

示例

下面的示例创建并激活了一个窗口。WOUTPUT() 可以显示此活动输出窗口的名称。关闭窗口并从内存中删除。如果另一个窗口是活动的，则显示它的名称。否则显示一条信息，指出输出是导向 Visual FoxPro 主窗口的。

```
DEFINE WINDOW wOutput1 FROM 2,2 TO 12,32 TITLE 'Output Window'  
ACTIVATE WINDOW wOutput1  
WAIT WINDOW 'wOutput1 window: ' + WOUTPUT( )  
RELEASE WINDOW wOutput1  
IF EMPTY(WOUTPUT( ))  
    WAIT WINDOW 'Output being directed to the main Visual FoxPro window'  
ELSE  
    WAIT WINDOW 'Output window: ' + WOUTPUT( )  
ENDIF
```

请参阅

ACTIVATE WINDOW, DEFINE WINDOW, WEXIST(), WONTOP(),
WVISIBLE()

WPARENT () 函数

返回活动窗口或指定窗口的父窗口名。

语法

WPARENT([WindowName])

返回值类型

字符型

参数描述

WindowName

指定一个窗口，由 WPARENT() 返回其父窗口名。如果指定窗口没有父窗口，则 WPARENT() 返回空字符串。如果省略 *WindowName*，则该函数返回活动输出窗口的父窗口名。如果活动输出窗口没有父窗口，WPARENT() 返回空字符串。

说明

可以用 DEFINE WINDOW 命令创建窗口，并将其放置在父窗口之内，该子窗口和父窗口结合在一起。例如，在父窗口中创建并激活的子窗口不能移出父窗口，如果父窗口移动，子窗口也随着移动。

示例

下面的示例定义了一个父窗口和子窗口。然后用 WPARENT() 来确定谁是父窗口。

```
CLEAR ALL
CLEAR
DEFINE WINDOW wParent ;
    FROM 1,1 TO 20,20 ;
    TITLE 'wParent'  &&父窗口
ACTIVATE WINDOW wParent
DEFINE WINDOW wChild ;
    FROM 1,1 TO 10,10 ;
    TITLE 'wChild' ;
    IN WINDOW wParent &&子窗口
ACTIVATE WINDOW wChild
WAIT WINDOW 'The parent window is ' + WPARENT( )
RELEASE WINDOW wParent, wChild
```

请参阅

[ACTIVATE WINDOW](#), [DEFINE WINDOW](#), [WCHILD\(\)](#)

_WRAP 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。

WREAD () 函数

包含此项是为了提供向后兼容性。请使用“[表单设计器](#)”。

WriteExpression 方法

把表达式写到属性中。设计和运行时可用。

语法

```
Object.WriteExpression[cPropertyName, cExpression]
```

参数描述

cPropertyName

指定要写表达式的属性名。

cExpression

指定字符串，将其作为表达式写入属性。

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，项目挂接程序，关系，形状，微调，文本框，计时器，工具栏

请参阅

[ReadExpression 方法](#)

WriteMethod 方法

把指定的文本写到指定的方法中。只在设计时可用。

语法

Control.WriteMethod(MethodName, MethodText)

参数描述

MethodName

指定要写入文本的方法名。

MethodText

指定要写入的文本。

说明

该方法仅为已有的方法设置方法文本。如果对象中没有包含所指定的方法，Visual FoxPro 会产生错误。

应用于

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，形状，微调，文本框，计时器，工具栏

请参阅

[ReadExpression 方法](#)，[ReadMethod 方法](#)，[WriteExpression 方法](#)

ROWS () 函数

返回活动窗口或指定窗口中的行数。

语法

ROWS([WindowName])

返回值类型

数值型

参数描述

WindowName

指定要 ROWS() 返回行数的窗口。在 Visual FoxPro 中，也可以指定工具栏的名称。

在 Visual FoxPro 中，由 ROWS() 返回的值取决于为窗口所指定的字体。大多数字体可以按各种不同的大小显示，一些字体是按比例分隔的，行对应于当前字体的高度。有关详细内容，请参阅本手册前面部分的“字体概述”。

如果系统窗口已经打开，且可见或隐藏，则可以在 ROWS() 函数中指定系统窗口的名称（如 Command（命令窗口）、Data Session（数据集窗口）、Browse（浏览窗口）

等)。如果所指定的系统窗口已经关闭，FoxPro 会产生错误信息，但调试窗口 (Debug) 例外。如果打开了调试窗口，不论是否可见、隐藏或已经关闭，都可以在 `WROWS()` 中包含它的名称。

也可以在 `WindowName` 中包括空字符串，以返回 Visual FoxPro 主窗口的行数。

注意 可以用空字符串在以下返回窗口位置与大小的函数中指定 Visual FoxPro 主窗口：`WLCOL()`，`WLROW()` 和 `WCOLS()` 等。

说明

如果不指定窗口名，则函数返回活动输出窗口中的行数；若没有活动窗口，则返回 Visual FoxPro 主窗口的行数。

请参阅

[SCOLS\(\)](#)，[SROWS\(\)](#)，[WCOLS\(\)](#)，[WLCOL\(\)](#)，[WLROW\(\)](#)

WTITLE () 函数

返回活动窗口或指定窗口的标题。

语法

WTITLE([WindowName])

返回值类型

字符型

参数描述

WindowName

指定用 WTITLE() 返回标题的窗口。如果包含了由 DEFINE WINDOW 命令创建的用户自定义的窗口名，则返回用 TITLE 子句指定给窗口的标题。

当发出 BROWSE WINDOW 命令，在一个用户自定义窗口中打开浏览窗口时，如果浏览窗口没有标题，WTITLE() 返回用户自定义窗口的标题；如果浏览窗口有标题，WTITLE() 返回其标题。

也可以在 *WindowName* 中使用空字符串，指定 Visual FoxPro 主窗口。

说明

可以用 WTITLE() 返回窗口标题，并在窗口上边显示该标题。如果省略 *WindowName*，则返回活动窗口的标题。如果调试窗口、跟踪窗口或命令窗口为活动窗口，或者输出定向到 Visual FoxPro 主窗口，则 WTITLE() 返回空字符串。

窗口名及标题 用户自定义窗口、系统窗口、工具栏（Visual FoxPro 的工具栏）和浏览窗口以如下方式命名：

当使用 DEFINE WINDOW 命令创建用户自定义窗口时，可以为它指定名称。用户自

定义窗口名和标题是不同的。默认情况下，用户自定义窗口没有标题。在创建窗口时若包括了 TITLE 子句，所指定的标题显示在窗口的标题栏，但这不是窗口名。

默认情况下，系统窗口是 Visual FoxPro 界面的一部分，它的名称与窗口标题相同。例如，系统窗口包括命令窗口 (Command)、查看窗口 (View) 和跟踪窗口 (Trace) 等。在 Visual FoxPro 中，每个工具栏的名称都与其标题相同。

程序窗口、编辑窗口、标签设计器与报表设计器窗口的名称都与所创建或修改的文件名相同。

浏览窗口的名称与其窗口标题相同，该窗口名及标题按以下三种方式之一指定：第一种是为表别名指定的默认标题，第二种是窗口标题（如果存在），第三种是“浏览”标题（如果存在）。

在默认情况下，浏览窗口名为表别名。

为了在命令和函数中指定系统窗口和工具栏的名称，可用引号将整个系统窗口或工具栏的名称括起来。例如，要在 Visual FoxPro 中隐藏“报表控制”工具栏，可发出以下命令：

```
HIDE WINDOW '报表控制'
```

窗口名的附加注释 如果您记不得指定给窗口的名称，可检查“窗口”菜单，所有窗口名都列在“窗口”菜单的底部。

如果发出 `BROWSE WINDOW WindowName` 命令，会出现两个窗口，其中浏览窗口是一个单独的窗口，并且呈现出所指定的用户自定义窗口的属性。

如果发出 `BROWSE` 命令，但没有包括 `WINDOW` 子句，而这时有活动的窗口，则浏览窗口具有该活动窗口的属性。可以在 `BROWSE` 命令中包括 `NORMAL` 来改变这种行

为。

可以在接受窗口名的命令和函数中包括带有空格的窗口名。例如 MOVE WINDOW, DEACTIVATE WINDOW 和 WONTOP()。包括的窗口名从第一个非空格字符开始,并一直指定到最后一个非空格字符为止。

例如,名为 Invoice Entry 的浏览窗口可以用此命令移动:

```
MOVE WINDOW invoice BY 1,1
```

用户自定义窗口名不能包含有空格,但浏览窗口和系统窗口名可以包含空格。

请参阅

[ACTIVATE WINDOW](#), [DEFINE WINDOW](#), [WEXIST\(\)](#), [WLAST\(\)](#),
[WONTOP\(\)](#), [WOUTPUT\(\)](#), [WREAD\(\)](#)

W VISIBLE () 函数

确定指定窗口是否激活并且没有隐藏。

语法

```
W VISIBLE(WindowName)
```

返回值类型

逻辑型

参数描述

WindowName

指定 WVISIBLE() 要处理的窗口的名称。在 Visual FoxPro 中，也可以指定工具栏的名称。还可以在 *WindowName* 中包括空字符串，来指定 Visual FoxPro 主窗口。

说明

如果已经显示或激活指定窗口，并且没有隐藏，则 WVISIBLE() 返回“真”(.T.)。可使用 SHOW WINDOW 和 ACTIVATE WINDOW 两条命令显示和激活窗口。

如果没有激活窗口，或用 HIDE WINDOW 命令隐藏了窗口，或用 DEACTIVATE WINDOW 命令使窗口不活动，或者窗口不存在，则 WVISIBLE() 返回“假”(.F.)。

请参阅

[ACTIVATE WINDOW](#), [DEFINE WINDOW](#), [WEXIST\(\)](#), [WONTOP\(\)](#),
[WOUTPUT\(\)](#)

YEAR () 函数

从指定的日期表达式中返回年份。

语法

YEAR(dExpression | tExpression)

返回值类型

数值型

参数描述

dExpression

指定的日期表达式，YEAR() 从该表达式返回年份。参数 *dExpression* 可以是返回值，可以为日期或日期型变量、数组元素或字段的函数；也可以是日期字符串，例如 {06/06/98}。

tExpression

指定日期时间表达式。YEAR() 从该表达式返回年份。

说明

YEAR() 总是返回带世纪数的年份，CENTURY 的设置 (ON 或 OFF) 并不影响此返回值。

示例

```
CLEAR  
? YEAR (DATE ( ))
```

请参阅

[DATE\(\)](#) , [DATETIME\(\)](#) , [DOW\(\)](#) , [SET CENTURY](#)

ZAP 命令

从表中删除所有记录，只留下表的结构。

语法

```
ZAP
```

```
[IN nWorkArea | cTableAlias]
```

参数描述

IN nWorkArea

指定要删除所有记录的表所在的工作区。

IN cTableAlias

指定要删除所有记录的表别名。

如果省略 *nWorkArea* 和 *cTableAlias*，则删除当前所选的工作区中表的所有记录。

说明

ZAP 命令等价于 DELETE ALL 和 PACK 联用，但 ZAP 速度更快。

如果 SET SAFETY 为 ON，Visual FoxPro 会提示是否要从当前表中删除记录。

发出 ZAP 命令，不会引发 DELETE 触发器。有关为表创建触发器的详细内容，请参阅 CREATE TRIGGER。

警告 用 ZAP 命令从当前表中删除的记录不可恢复。

请参阅

[CREATE TRIGGER](#), [DELETE](#), [PACK](#)

ZoomBox 属性

指定一个表单是否具有缩放框。设计时可用，运行时可读写。

语法

Object.ZoomBox = lExpr

参数描述

lExpr

ZoomBox 属性的设置为：

设置

说明

“真”

（默认值）在表单的右上角显示一个缩放框。

(.T.)

“假”

在表单上不显示缩放框。

(.F.)

应用于

表单

请参阅

[BorderStyle](#) 属性，[SizeBox](#) 属性

ZOOM WINDOW 命令

更改用户自定义窗口或 Visual FoxPro 系统窗口的大小与位置。

语法

ZOOM WINDOW WindowName MIN | MAX | NORM

```
[AT nRow1, nColumn1 | FROM AT nRow1, nColumn1  
[SIZE AT nRow2, nColumn2 | TO nRow2, nColumn2]]
```

参数描述

WindowName

指定要更改大小的窗口的名称。

MIN

将窗口缩小到最小尺寸。

在 Visual FoxPro 中，窗口缩为一个图标。

所有系统窗口可以缩小到最小尺寸。Visual FoxPro 系统窗口在最小化之前，必须先要在 Visual FoxPro 主窗口或用户自定义窗口中打开它。

用户自定义窗口在定义之后可以最小化，且在更改大小之前不必激活它。

MAX

使窗口扩大到充满整个 Visual FoxPro 主窗口、Windows 桌面、Macintosh 桌面或用户自定义窗口。如果子窗口放置在父窗口之中，当子窗口最大化时，将充满父窗口。如果有任何同 MAX 在一起使用的附加 ZOOM WINDOW 子句（如 AT, SIZE, TO 或 FROM 等），则忽略 MAX。

只有用 ZOOM 命令定义的用户自定义窗口可以扩大到最大尺寸。

NORM

在窗口最小化或最大化之后，将窗口返回到其原来大小。也可以用 NORM 移动窗口，而不更改其大小。可以用不带有其他附加子句的 ZOOM WINDOW NORM 命令，将最大化或最小化的窗口还原成原来的大小和位

置。

AT *nRow1*, *nColumn1* | FROM *nRow2*, *nColumn2*

可以包含 AT 或 FROM 子句，来指定窗口的位置。

ZOOM WINDOW *WindowName* NORM AT *nRow1*, *nColumn1* 命令将最大化或最小化的窗口恢复到其原来的大小，并放置在指定的位置上。窗口左上角的位置由坐标 AT *nRow1*, *nColumn1* 指定。使用 MOVE WINDOW 可以更改窗口位置。

若包含了 NORM 子句，窗口左上角则放置在由 AT *nRow1*, *nColumn1* 指定的位置，该位置是在 Visual FoxPro 主窗口或 FoxPro for Windows 窗口中。如果包含了 MIN，则忽略 AT 和 FROM，且窗口显示为一个图标，该图标位于 Visual FoxPro 主窗口或 FoxPro for Windows 窗口中的下部。如果包含了 MAX，则忽略 AT 和 FROM，且将窗口扩大到充满 Visual FoxPro 主窗口或 FoxPro for Windows 窗口。

如果用 IN DESKTOP 子句创建窗口，则窗口的左上角放在由 AT *nRow1*, *nColumn1* 指定的位置，该位置是在 Windows 桌面上。如果包含了 MIN，则忽略 AT 和 FROM，且窗口显示为 Windows 桌面下部的一个图标。如果包含了 MAX，则忽略 AT 和 FROM，且将窗口扩大到充满 Windows 桌面。

SIZE AT *nRow2*, *nColumn2* | TO *nRow2*, *nColumn2*

也可以包括 SIZE 或 TO 子句来指定窗口的大小。如果包含 SIZE，窗口大小即高为 *nRow2* 行，宽为 *nColumn2* 列。如果包含了 TO，窗口左上角保持在当前位置，窗口右下角放置在由 *nRow2*, *nColumn2* 指定的位置。

说明

可以将窗口缩小到最小尺寸，或者扩大窗口充满整个 Visual FoxPro 主窗口，或者将窗

口缩放到其间任一大小。

在 Visual FoxPro for Windows 中，如果使用 DEFINE WINDOW 和 IN DESKTOP 命令创建用户自定义窗口，则所创建的窗口可以扩大到充满整个 Windows 桌面。

窗口可以由最小直接放到最大，或者相反。

还可以指定已调整的窗口在 Visual FoxPro 主窗口或用户自定义窗口中的位置。

为了缩放系统窗口，可以用引号将整个系统窗口名引起来，例如，要使命令窗口最小化，可发布以下命令：

```
ZOOM WINDOW 'Command Window' MAX
```

所有系统窗口都可以用 ZOOM WINDOW 命令来调整大小。

示例

下面的示例为表 customer 打开了一个“浏览”窗口。最小化“浏览”窗口。“浏览”窗口恢复到默认大小。在特定的位置中再次最小化“浏览”窗口。然后放大“浏览”窗口到特定的大小，直到最大化。

```
CLEAR ALL
CLEAR
CLOSE DATABASES
OPEN DATABASE (HOME(2) + 'Data\Testdata')
USE customer && 打开 Customer 表
BROWSE NORMAL NOWAIT
IF _DOS OR _WINDOWS
    ZOOM WINDOW customer MIN
    WAIT WINDOW TIMEOUT 3 ;
    'MIN clause - This window will timeout. Please wait.'
```

```
ENDIF
ZOOM WINDOW customer NORM
WAIT WINDOW TIMEOUT 3 ;
    'NORM clause - This window will timeout. Please wait.'
IF _DOS OR _WINDOWS
    ZOOM WINDOW customer MIN AT 10,10
    WAIT WINDOW TIMEOUT 3 ;
        'MIN AT 10,10 clause - This window will timeout. Please wait.'
ENDIF
ZOOM WINDOW customer NORM AT 1,1 SIZE 22,25
WAIT WINDOW TIMEOUT 3 ;
    'NORM & SIZE clauses - This window will timeout. Please wait.'
ZOOM WINDOW customer NORM FROM 10,10 TO 22,70
WAIT WINDOW TIMEOUT 3 ;
    'NORM & TO clauses - This window will timeout. Please wait.'
ZOOM WINDOW customer MAX
WAIT WINDOW TIMEOUT 3 'MAX clause - This window will timeout. Please wait.'
CLEAR ALL
```

请参阅

ACTIVATE WINDOW , DEFINE WINDOW , SIZE WINDOW

ZOrder 方法

将一个指定的表单或控件在图形层次放在 z-order 的前面或后面。将一个包含在工具栏对象中的控件放在控件数组的前面或后面，可以确定该控件在工具栏中显示的顺序。

语法

```
[Object.]ZOrder([nOrder])
```

参数描述

nOrder

指定一个整数，表明一个对象相对于其他对象的位置。如果省略 nOrder，则设置为 0。

设置

说明

0 (默认值) 对象位于 z-order 的前面。

1 对象位于 z-order 的后面。

说明

有两种图形层次与对象相关。后层是绘制空间，在这里显示图形方法的结果，而前层是对象层。一层的内容遮挡后面一层的内容。ZOrder 方法只在显示对象的层调整对象。

注意 对于页面对象，ZOrder 方法不影响 PageOrder 属性的设置。ZOrder 只确定顶层的活动页面。

有关使用控件的详细内容，请参阅《Microsoft Visual FoxPro 6.0 中文版程序员指南》的第十章“使用控件”。

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，OLE 绑定型控件，OLE 容器控件，选项按钮，选项按钮组，页面，页框，屏幕，形状，微调，文本框，工具栏

请参阅

[PageOrder 属性方法](#)

