软件工程师丛书

Visual FoxPro 6.0/7.0 命令与函数大全

郝 锋 何曙光 等编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

831
TP311.138F0
H2 4

软件工程师丛书

Visual FoxPro 6.0/7.0 命令与函数大全

郝 锋 何曙光 等编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 提 要

Visual FoxPro 7.0 是一个功能强大的面向对象的关系数据库管理系统，是 Microsoft 公司最新推出的可视化开发工具的组成部分，本书详尽地介绍了 Visual FoxPro 6.0/7.0 的全部命令、函数及系统变量等内容。全书分为命令和函数两部分，按命令或函数的用途分章介绍各命令或函数的语法、用途、参数描述和使用等，并对一些重要的命令或函数提供了完整范例。书中对 Visual FoxPro 7.0 增强型或新增的命令和函数给出了明确的标注。

本书内容十分丰富，是 Visual FoxPro 6.0/7.0 应用开发者必不可少的工具书，也可作为 FoxPro 的教学参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Visual FoxPro 6.0/7.0 命令与函数大全 / 郝锋等编著. —北京：电子工业出版社，2002.3
（软件工程师丛书）
ISBN 7-5053-7524-5

Ⅰ.V... Ⅱ.郝... Ⅲ.关系数据库－数据库管理系统，Visual FoxPro 6.0/7.0 Ⅳ.TP311.138

中国版本图书馆 CIP 数据核字（2002）第 014736 号

责任编辑：段来盛
印　　刷：北京市天竺颖华印刷厂
出版发行：电子工业出版社　http://www.phei.com.cn
　　北京市海淀区万寿路 173 信箱　邮编　100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：44.25　字数：1120 千字
版　　次：2002 年 3 月第 1 版　2002 年 3 月第 1 次印刷
印　　数：4000 册　　定价：66.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

目录

第一部分 命 令

第二部分　函　　数

此为试读,需要完整PDF请访问: www.ertongbook.com

第一部分　命　令

本部分分类介绍了 Visual Foxpro 6.0/7.0 的所有命令，对每条命令分别给出了其语法格式、用途、参数描述、使用说明及相关命令和函数，对大部分的命令还给出了使用范例。

1

数据库操作命令

ADD TABLE 命令

【语法】

```
ADD TABLE TableName | ?
[NAME LongTableName]
```

【用途】将一个指定表添加到当前数据库中。

【参数描述】

TableName：为表名，指定要添加到当前数据库中的表。

?：显示“打开”文件对话框，从磁盘上选择某个数据表(.DBF 文件)，添加到当前的数据库中。

NAME LongTableName：**LongTableName** 为要添加到数据库中的表的长名。长名是表在数据库中的表名，最多可以包含 128 个字符。若不指定表的长名，则在数据库中使用表的文件名作为表名。注意，长名只在数据库中有效，自由表无长名。

【使用说明】当一个表添加到数据库以后，就可以利用数据库的特性操作该表。一个表一旦添加到数据库后就不再是自由表。如果需要，可以使用 **REMOVE TABLE** 命令将指定数据库中的任何一个表移出数据库，使其变为自由表。

要添加表到数据库中，必须具备下列条件：

- 表是有效的.DBF 文件。
- 除非为添加的表指定了惟一的长名，否则该表不能与打开数据库中已有的表同名。
- 表必须是自由表，不能在另一个数据库中。
- 必须以独占方式打开数据库。
- 不包含在事务中。

【范例】使用 **CREATE DATABASE** 命令创建名为“通信录 1”和“通信录 2”两个数据库，用 **SET DATABASE TO** 命令指定“通信录 1”为当前数据库。在数据库“通信录 1”中创建名为“朋友 1”的表。用 **DISPLAY DATABASE** 查看当前数据库信息，然后关闭“朋友 1”表，并用 **REMOVE TABLE** 将其从数据库“通信录 1”中移出。接着，设定“通信录 2”为当前数据库，使用 **ADD TABLE** 命令将“朋友 1”表添加到数据库“通信录 2”中。然后，使用 **RENAME** 命令将数据库中的表名“朋友 1”改为“朋友 2”。最后用 **DELETE DATABASE** 命令删除这两个数据库及其中的表。

```
CLOSE DATABASE
CREATE DATABASE 通信录 1                    &&创建“通信录 1”数据库
CREATE DATABASE 通信录 2                    &&创建“通信录 2”数据库
SET DATABASE TO 通信录 1                    &&指定当前数据库为“通信录 1”
CREATE TABLE 朋友 1 (姓名 C(20), 电话 C(20))  &&创建“朋友 1”表
CLEAR
WAIT "按任意键显示当前数据库信息"
DISPLAY DATABASE                            &&显示当前数据库信息
CLOSE TABLES                                &&关闭数据库中的表
REMOVE TABLE 朋友 1                         &&从当前数据库中移出“朋友 1”表
DISPLAY DATABASE
SET DATABASE TO 通信录 2                    &&指定当前数据库为“通信录 2”
ADD TABLE 朋友 1                            &&将“朋友 1”表添加到当前数据库中
RENAME TABLE 朋友 1 TO 朋友 2               &&更改数据库表名
WAIT "按任意键显示通信录 2"
DISPLAY DATABASE                            &&显示当前数据库信息
CLOSE DATABASE ALL
DELETE DATABASE  通信录 1 DELETETABLES &&删除数据库，及其所含表
DELETE DATABASE  通信录 2 DELETETABLES
```

【相关命令和函数】CLOSE DATABASES，CREATE DATABASE，DISPLAY TABLES，FREE TABLE，

OPEN DATABASE，REMOVE TABLE。

APPEND 命令

【语法】

```
APPEND [BLANK]
[IN nWorkArea | cTableAlias]
[NOMENU]
```

【用途】在指定表中追加一个或多个新记录。

【参数描述】

BLANK：在当前表的末尾添加一个空记录。注意，在发出 APPEND BLANK 命令时并不会自动打开编辑窗口。可以在 BROWSE、CHANGE 或 EDIT 窗口中编辑新记录。

IN nWorkArea：nWorkArea 为工作区号，指定追加新记录表所在的工作区。

IN cTableAlias：cTableAlias 为表的别名，指定追加新记录表的别名。

NOMENU：将“表”菜单标题从系统菜单栏中删除，以避免通过这个菜单改变编辑窗口的格式。

【使用说明】当发出 APPEND 或 APPEND BLANK 命令时，如果在当前选定工作区中没有打开的表，将显示一个“打开”文件对话框，以便从磁盘上选择需要追加记录的表。执行 APPEND 命令后，在打开的编辑窗口中，可以输入一个或多个新记录。增加新记录后，Visual FoxPro 将自动修改打开表的所有索引。

如果省略 nWorkArea 或 cTableAlias，将追加新记录到当前选定工作区的表中。

如果发出的 APPEND 命令带 nWorkArea 或 cTableAlias 参数，则空记录将添加到由 nWorkArea 或 cTableAlias 指定的工作区的表中，并且自动选定该表。

如果发出的 APPEND BLANK 命令带 nWorkArea 或 cTableAlias 参数，则空记录将添加到指定的 nWorkarea 或 cTableAlias 工作区的表中，但不选定该表。

【范例】创建一个名为 test.dbf 的表，该表只包含一个数值字段 xh。利用循环和 APPEND [BLANK]命令为 test.dbf 表添加 10 个记录，每个记录的 xh 字段填入序号。最后列出全表。

```
CLOSE DATABASES                  &&关闭数据库
CREATE TABLE test (xh N(5))      &&创建自由表
FOR i = 1 TO 10                  &&循环 10 次增加 10 个记录
APPEND BLANK                     &&产生新记录
REPLACE xh WITH i+1000           &&在 xh 字段中填入序号
ENDFOR                           &&循环结束
CLEAR                            &&清主窗口
LIST                             &&在主窗口显示全表
```

【相关命令和函数】APPEND FROM ARRAY，BROWSE，CHANGE，EDIT，INSERT-SQL，REPLACE。

APPEND FROM 命令

【语法】

```
APPEND FROM FileName | ?
[FIELDS FieldList]
[FOR lExpression]
[[TYPE] [DELIMITED [WITH Delimiter | WITH BLANK | WITH TAB
| WITH CHARACTER Delimiter]
| DIF | FW2 | MOD | PDOX | RPD | SDF | SYLK
| WK1 | WK3 | WKS | WR1 | WRK | CVS
| XLS | XL5 [SHEET cSheetName] | XL8 [SHEET cSheetName]]]
[AS nCodePage]
```

【用途】从指定数据文件(Visual FoxPro 表或其他软件的数据文件)中读取数据，并追加数据到当前表中

来。

【参数描述】

FileName：文件名，指定读取数据的文件。如果给出的文件名不包含扩展名，则将其默认为 Visual FoxPro 扩展名(.DBF)。如果文件是 Visual FoxPro 表，则无论 SET DELETED 命令为何种设置，表中带删除标记的记录也将追加到当前表中。

?：显示“打开”文件对话框，从中选择要读取的数据文件。

FIELDS FieldList：FieldList 为用逗号(,)分隔的字段名列表，指出当前表要追加数据的字段。

FOR lExpression：lExpression 为一逻辑表达式，作为筛选条件，当其为“真”(.T.)时，读取文件中的数据追加到当前表中。如果省略 FOR，则文件数据都将追加到当前表中。

TYPE：指定读取数据的文件类型。如果该文件类型不是 Visual FoxPro 表，则必须指定其类型。指定类型时，不必包括 TYPE 关键字。可从各种类型文件(包括分隔 ASCII 文本文件)中读入数据添加到当前表中。可以指定这些文件中数据字段的分隔符。

如果要追加的数据文件扩展名不是默认的扩展名(.DBF)，则源文件名必须包括文件扩展名。例如，对于 Microsoft Excel 工作表通常要指定其扩展名为.XLS。注意，如果要追加的记录来自工作表，工作表中的数据必须以主行序而非主列序存储，这样才能使追加的工作表数据符合 Visual FoxPro 的表结构。

DELIMITED：指定源文件为分隔数据文件。分隔数据文件就是 ASCII 文本文件，其扩展名默认为 .TXT 。在这类文件中，每行(记录)以“回车”和“换行”符结尾。各字段内容默认情况是由逗号(,)分开，字符字段的数据还需要用引号(“”)引起来。例如：

```
"李明",8564328,"188-53518"
```

如果文本文件中的日期格式正确，还可以从中导入日期数据，默认格式为 mm/dd/yy。还可以选择加入世纪信息。Visual FoxPro 导入日期数据(如 12/25/95)不包含世纪信息，世纪信息的默认值为 20 世纪。日期分隔符可为任意非数值字符，但不能使用文件中的字段分隔符。

如果其他一些日期格式与 SET DATE 中指定的格式相匹配，Visual FoxPro 也可导入这些格式的日期数据。若要导入非默认格式的日期，应在使用 APPEND FROM 命令前，先发出 SET DATE 命令修改日期数据格式设置。要想检查日期格式是否能成功地导入，可使用 CTOD()函数。如果 CTOD()函数接收此日期值，则日期数据可正确地导入。

DELIMITED WITH Delimiter：Delimiter 为一字符。指定文本文件中的字符字段的数据标识是 Delimiter，而不是引号。

DELIMITED WITH BLANK：指定文本文件中由空格符 (BLANK) 作为字段分隔符，而不是用逗号分隔字段。

DELIMITED WITH TAB：指定文本文件中以制表符(TAB)作为字段分隔符，而非逗号。

DELIMITED WITH CHARACTER Delimiter：Delimiter 为一字符。指定在文本文件中，由 Delimiter 为字段之间的分隔符，而不是逗号。如果 Delimiter 是分号，应该用引号引起来，因为分号在 Visual FoxPro 程序中起续行符作用。Delimiter 也可以是 BLANK 或 TAB。

WITH Delimiter 子句可以与 WITH CHARACTER Delimiter 子句同时使用。例如，在下面的例子中，读取一个文本文件数据。该文本文件中，字符字段分隔符使用下划线(_)，而字段之间用星号(*)作为分隔符。

```
APPEND FROM mytxt.txt DELIMITED WITH _  WITH CHARACTER *
```

DIF：选用 DIF 可以从 VisiCalc .dif(数据交换格式)文件中导入数据。矢量(列)对应当前选定表的字段，元组(行)对应表的记录。DIF 文件的默认扩展名为.DIF。

FW2：选用 FW2 可以从由 Framework II 创建的文件中导入数据。FW2 文件的默认扩展名为.FW2。

MOD：选用 MOD 可以从 Microsoft Multiplan 4.01 版本的文件中导入数据。MOD 文件由 Microsoft Multiplan 4.01 版本创建，默认扩展名为.MOD。

PDOX：选用 PDOX 可以从 Paradox 3.5 版或 4.0 版数据库文件中导入数据。Paradox 文件名的默认扩展名为.DB。

RPD：选用 RPD 可以从由 RapidFile 1.2 版本创建的文件中导入数据。RapidFile 文件名的默认扩展名为.RPD。

此为试读，需要完整PDF请访问：www.ertongbook.com

SDF：选用 SDF 可以从系统数据格式文件中导入数据。SDF 文件是一种 ASCII 文本文件，记录长度固定，并且以回车和换行符结尾，各字段不分隔开。文件的默认扩展名为.TXT。

SYLK：选用 SYLK 可以从 SYLK(符号链接)交换格式文件中导入数据。SYLK 文件用于 Microsoft MultiPlan 中。SYLK 文件中的列对应 Visual FoxPro 表的字段，行对应表的记录。SYLK 文件没有扩展名。

WK1：选用 WK1 可以从 Lotus1-2-3 2.x 版本的电子表格中导入数据。电子表格的每列为表的一个字段，每行为表的一条记录。Lotus1-2-3 2.x 版本创建的电子表格扩展名为 WK1。

WK3：选用 WK3 可以从 Lotus1-2-3 的电子表格中导入数据，电子表格的每列为表的一个字段，每行为表的一条记录。Lotus1-2-3 版本 3.X 创建的电子表格扩展名为.WK3。

WKS：选用 WKS 可以从 Lotus1-2-3 1-A 版的电子表格中导入数据。电子表格的每列为表的一个字段，每行为表的一条记录。Lotus1-2-3 1-A 版本创建的文件扩展名为.WKS。

WR1：选用 WR1 可以从 Lotus Symphony 1.1 或 1.2 版的电子表格中导入数据。电子表格的每列为表的一个字段，每行为表的一条记录。Symphony 1.1 或 1.2 版创建的电子表格扩展名为.WR1。

WRK：选用 WRK 可以从 Lotus Symphony 1.0 版的电子表格中导入数据。电子表格中的每列为表的一个字段，每行为表的一条记录。Symphony 1.0 版创建的电子表格扩展名为.WRK。

CVS：选用 CVS 可以从一个各值用逗号分隔的文件中导入数据。CSV 文件的第 1 行是字段名，导入该文件时将忽略这个字段名。

XLS：选用 XLS 可以从 Microsoft Excel 工作表中导入数据。工作表的每列为表的一个字段，每行为表的一条记录。由 Microsoft Excel 创建的工作表扩展名为.XLS。

XL5：选用 XL5 可以从 Microsoft Excel 5.0 版中导入数据。工作表的每列为表的一个字段，每行为表的一条记录。工作表文件的扩展名为.XLS。如果省略 SHEET 子句，则导入 Sheet1 中的数据。为了导入特定工作表中的数据，需要包含 SHEET 关键字，并且使用 cSheetName 指定工作表的名称。

XL8：包含 XL8 可以导入 Microsoft Excel 97 的数据。工作表的列变成表中的字段，工作表的行变成表中的记录。在 Microsoft Excel 中创建的工作表文件的扩展名是 .xls。如果省略 SHEET 子句，会导入 Sheet1 中的数据。为了导入特定工作表中的数据，需要包含 SHEET 关键字，并且使用 cSheetName 指定工作表的名称。

AS nCodePage：指定源表或源文件的代码页。Visual FoxPro 将复制源表或源文件的内容，并在复制时自动将数据转换到当前表的代码页中。

如果指定的 nCodePage 值无法使用，Visual FoxPro 将产生一条错误信息。可以用 GETCP()函数显示代码页对话框，在对话框中可以为追加的表或文件指定代码页。如果省略 AS nCodePage 子句，并且 Visual FoxPro 不能判定源表或文件的代码页，Visual FoxPro 将复制源表或文件内容，并在复制数据的过程中，自动将数据转换到当前的 Visual FoxPro 代码页中。如果 SET CPDIALOG 为 ON，当前选定工作区中的表以代码页标记。如果要从没有代码页标记的表中读入数据并添加到表中时，将显示代码页对话框，可以在其中选择表的代码页。当前的 Visual FoxPro 代码页可以由 CPCURRENT()函数设定。

如果省略 AS nCodePage 子句，并且 Visual FoxPro 可确定追加记录的表或文件的代码页，Visual FoxPro 将复制表或文件的内容，并在复制数据的过程中，自动将数据转换到当前选定表的代码页中。

如果 nCodePage 为 0，Visual FoxPro 认为需追加记录的表和文件的代码页与当前选定表的代码页相同，并且不转换代码页。

【使用说明】如果读取数据源文件是 Visual FoxPro 表或在 FoxPro 早期版本中创建的表，则其扩展名默认为 .DBF。如果其扩展名不是 .DBF，则必须指定其扩展名。如果文件不是 Visual FoxPro 表或 FoxPro 早期版本创建的表，则必须指定文件的类型。

在读取 DBASE IV 或 DBASE V 创建的包含备注字段的表之前，必须先用 USE 命令在 Visual FoxPro 中打开该表，当提示信息询问是否要转换文件时，请选择“是”。

如果是从 FoxPro 早期版本的表或 Visual FoxPro 表中读取数据，则该表可以在另一工作区打开。对该表中有删除标记的记录，在添加到当前表中后删除标记丢失。

使用 DBF()函数可以从一个只读的临时表追加数据，该临时表使用 SELECT - SQL 命令创建。如下例所示，使用 DBF()函数读取表的数据：

```
APPEND FROM DBF('<Cursor Name>')
```

【范例】创建和编辑文本文件 test.txt，录入若干行姓名和电话号码数据，以逗号分隔，用双引号区分字符串。创建一个名为 Mytable1 的表，包括“姓名”和“电话”两个字段；用 APPEND FROM 命令将 test.txt 中的内容导入到数据表 Mytable1.DBF 中。

```
CLEAR                                                    &&清窗口
CREATE TABLE Mytable1 (姓名 C(10), 电话 C(10) )          &&创建表
* 编辑文本，输入几行数据，姓名和电话号码用逗号分隔，字符串用双引号。
MODIFY FILE test.txt                                     &&创建和编辑文本文件
APPEND FROM test.txt FIELDS 姓名,电话 DELIMITED          &&导入文本文件中的数据
LIST                                                     &&显示表的记录数据
```

【相关命令和函数】COPY FILE，COPY TO，EXPORT，GETCP()，IMPORT。

APPEND FROM ARRAY 命令

【语法】

```
APPEND FROM ARRAY ArrayName
[FOR lExpression]
[FIELDS FieldList
| FIELDS LIKE Skeleton
| FIELDS EXCEPT Skeleton]
```

【用途】将指定数组的每一行数据当作一条记录追加到当前表中。

【参数描述】

ArrayName：ArrayName 为数组名，该数组包含要追加到当前表里的数据。

FOR lExpression：lExpression1 为一逻辑表达式，作为筛选条件，此条件表达式中必须包含目标字段名。只有当这个表达式为“真”(.T.)时，数组中的对应行数据才追加到表中。

在数组某行数据追加到表中的记录之前，先检查与 lExpression 中指定的目标字段对应的数组元素是否满足 lExpression 条件。如果数组元素满足条件，则将对应行的数组数据作为一个记录追加到表中。如果数组元素不满足条件，则这一行数据不追加到表中，并继续检查下一行。

FIELDS FieldList：FieldList 为用逗号(，)分隔的字段名列表，其指定当前表中要从数组中追加数据的各字段。FieldList 中的第 1 个字段用数组第 1 个元素的内容更新，第 2 个字段用数组第 2 个元素更新，依此类推。

FIELDS LIKE Skeleton：Skeleton 为字段梗概，表中凡与此梗概匹配的字段都将从数组追加数据。字段梗概 Skeleton 支持通配符(？)和(*)。例如，指定当前表中所有以字母 B 与 W 开头的字段从数组 MyArray 追加数据，可使用：

```
APPEND FROM ARRAY MyArray FIELDS LIKE B*,W*
```

FIELDS EXCEPT Skeleton：指定从数组中更新当前表中所有与此字段梗概 Skeleton 不匹配的字段。凡是与此处 Skeleton 不匹配的字段都要从数组追加数据。LIKE 子句与 EXCEPT 子句可以组合使用。例如：

```
APPEND FROM ARRAY aMyArray FIELDS LIKE B*,P* EXCEPT PARTNO*
```

【使用说明】APPEND FROM ARRAY 命令将忽略表的备注字段和通用字段。如果表处于打开状态并处于共享使用中，则在追加记录时，APPEND FROM ARRAY 命令将锁定表头。

如果指定的是一维数组，则 APPEND FROM ARRAY 命令只在当前表中追加一个记录。数组的第 1 个元素数据填充到记录的第 1 个字段，第 2 个元素数据填充到记录的第 2 字段，依此类推。如果一维数组的元素个数多于表的字段数，则忽略多余的数组元素。如果表的字段数多于数组元素个数，则表的多出字段将初始化为字段默认值。

如果指定的是二维数组，则 APPEND FROM ARRAY 命令将数组的每一行数据作为一个记录追加到当前表中。数组的多行将对应多条记录。例如，如果数组有 10 行，则在表中追加 10 个新记录。数组的第 1

列数据赋值给新添加记录的第 1 个字段，第 2 列数据赋值给新记录的第 2 个字段，依此类推。假定数组有四行三列，则数组中的第 1 列元素分别赋值给四个新记录的第 1 个字段。如果二维数组的列数多于表的字段数，则将忽略多余的列。如果表字段数多于数组列数，则多出的字段将初始化为字段默认值。

如果数组数据与相应字段数据类型兼容，那么即使相应的数组元素的数据类型与字段数据类型不匹配，APPEND FROM ARRAY 也能填充字段。如果数据不兼容，则将初始化字段为字段默认值。

下面给出了各种字段类型对应的默认空值：

字段类型	默认值
字符型	空格
数值型	0
货币型	0
浮点型	0
整型	0
双精度型	0
日期型	空日期(例如， CTOD(''))
日期时间型	空日期时间(例如， CTOT(''))
逻辑型	假(.F.)
备注型	空(无内容)

【范例】定义一个一维数组 person，为其赋值；创建一个新表 MyTable，并用 APPEND FROM ARRAY 命令将 person 数组中的数据添加到表 MyTable 中来。

```
LOCAL ARRAY person(2)          &&定义一个一维数组
person(1)="李明"               &&为数组第 1 个元素赋值
person(2)="884679"             &&为数组第 2 个元素赋值
CREATE TABLE MyTable (姓名 C(10), 电话 C(10) )
APPEND FROM ARRAY person       &&将 person 数组的内容加入当前表
LIST                           &&显示表记录数据
```

【相关命令和函数】APPEND，COPY TO ARRAY，DIMENSION，GATHER，SCATTER。

APPEND GENERAL 命令

【语法】

```
APPEND GENERAL GeneralFieldName
[FROM FileName]
[DATA cExpression]
[LINK]
[CLASS OLEClassName]
```

【用途】从包含 OLE 对象的文件中导入 OLE 对象到当前表的指定通用字段中。

【参数描述】

GeneralFieldName：当前表中放置 OLE 对象的通用字段名。GeneralFieldName 可以用带有表别名的方式来指定在非当前工作区中打开表的通用字段。

FROMFileName：FileName 为文件名，该文件中包含了 OLE 对象。必须给出包括扩展名的文件全名。如果文件不在当前目录中，还需要给出文件的路径。

DATAcExpression：cExpression 为一字符表达式，计算后以字符串形式传递给通用字段中保存的 OLE 对象。该 OLE 对象必须能接收和处理字符串。例如，不能在 Paintbrush 的图片对象中存入字符串。

LINK：建立 OLE 对象和包含该对象的文件间的链接。LINK 参数使 OLE 对象出现在通用字段中，但对象定义仍在文件中。如果省略 LINK，OLE 对象将嵌入到通用字段中。

CLASS OLEClassName：为 OLE 对象指定具体的 OLE 类，而不用其默认类。OLEClassName 为指定的类名。注意，可以通过运行 REGEDIT 并双击某一 OLE 对象来确定该对象的类名，类名列在“标识符”后。当包含 OLE 对象的文件的扩展名不同于默认扩展名，并且要强制类行为时，可以指定类名。如果默认扩展名可以用于多个 OLE 服务程序，可以用该类指定具体的服务程序。

【使用说明】如果在表的指定通用字段中已有一个 OLE 对象，则这个 OLE 对象将由从文件中导入的 OLE 对象取代。若要从指定通用字段中删除一个 OLE 对象，可以不带任何附加参数执行 APPEND GENERAL GeneralFieldName 命令(GeneralFieldName 是要清理的通用字段的名称)。

【范例】创建一个表 MyGenTbl，其中只有一个通用字段 Mygenfield,用 APPEND GENERAL 命令将 C:\Windows\ Clouds .Bmp 图片导入 MyGenTbl 表的通用字段 Mygenfield 中。浏览当前表，在浏览窗口中用鼠标双击记录中通用字段 mygenfield 处的 Gen，即可以打开导入的对象。

```
CREATE TABLE MyGenTbl (Mygenfield G)      &&创建表
APPEND BLANK                              &&添加一个空记录
APPEND GENERAL Mygenfield FROM C:\Windows\Clouds.BMP
BROWSE
```

【相关命令和函数】@ ... SAY ， MODIFY GENERAL。

APPEND MEMO 命令

【语法】

```
APPEND MEMO MemoFieldName FROM FileName
[OVERWRITE] [AS nCodePage]
```

【用途】将指定文本文件中的内容复制到选定表当前记录的指定备注字段中。

【参数描述】

MEMO MemoFieldName：MemoFieldName 指定备注字段名，一个文本文件的内容将追加到此备注字段中。

FROM FileName：FileName 为文本文件名，该文件的其内容将被复制到备注字段中。FileName 必须给出包括扩展名的文件全名。如果文件不在当前目录中，还需要给出文件的路径。

OVERWRITE：指示用文件的内容替换备注字段当前的内容。如果忽略参数 OVERWRITE，则文本文件的全部内容将以追加方式追加到当前记录的指定备注字段中。

AS nCodePage：指定复制到备注字段中的文本文件的代码页。Visual FoxPro 复制文本文件的内容，并在将数据复制到备注字段的过程中，自动将数据从指定代码页转换成备注字段所在表的代码页。如果包含备注字段的表没有使用代码页标记，Visual FoxPro 自动将数据从指定代码页转换到当前 Visual FoxPro 代码页。如果指定的 nCodePage 值无效，Visual FoxPro 将产生一条错误信息。可以使用 GETCP()函数显示“代码页”对话框，并从中指定要追加的表或文件的代码页。如果忽略 AS nCodePage 子句，或者指定 nCodePage 值为 0，将不转换文本文件的代码页。

【范例】建立一个只有一个备注字段的表 Mytable；创建一个文本文件 a.txt，在该文本文件中输入所需要的信息。用 APPEND MEMO 命令将 a.txt 内容添加到 Mytable 表当前记录的备注字段 notes 中。用 LIST 命令显示表的 notes 备注字段。

```
CREATE TABLE Mytable (notes M)   &&创建表
MODIFY FILE a.txt                &&创建文本文件，并输入信息
APPEND BLANK
APPEND MEMO notes FROM a.txt     &&将 a.txt 文件内容加入到 notes 备注字段
DELETE FILE a.txt
LIST notes                       &&显示表的 notes 备注字段
```

【相关命令和函数】COPY MEMO，GETCP()。

APPEND PROCEDURES 命令

【语法】

```
APPEND PROCEDURES FROM FileName
[AS nCodePage] [OVERWRITE]
```

【用途】将指定文本文件中的存储过程追加到当前数据库中。

【参数描述】

FROM FileName：FileName 为文本文件名，该文本文件中的函数或过程将追加到当前数据库的内部存储过程中。FileName 必须给出包括扩展名的文件全名。如果文件不在当前目录中，还需要给出文件的路径。

AS nCodePage：指定要追加其内部存储过程的文本文件所在的代码页。Visual FoxPro 在复制文本文件的内容时，自动将文本文件的内容转换成指定的代码页。

如果指定的 cCodePage 值无效，Visual FoxPro 将产生错误信息。可以使用 GETCP() 函数显示“代码页”对话框，并从中指定文本文件的代码页，此文本文件包含要追加的内部存储过程。

如果忽略 AS nCodePage，Visual FoxPro 将复制文本文件的内容，并自动将文本文件内容转化成 Visual FoxPro 的当前代码页。可以用 CPCURRENT() 函数设置 Visual FoxPro 的当前代码页。

如果 nCodePage 值为 0，Visual FoxPro 将假定文本文件的代码页与当前数据库的代码页相同，不转换代码页。

OVERWRITE：指示用文本文件中的过程覆盖数据库中的当前内部存储过程。如果忽略 OVERWRITE 参数，则文本文件中的内部存储过程追加到当前数据库内部存储过程中。

【使用说明】本命令可以用编程方式修改数据库中的内部存储过程。执行 APPEND PROCEDURES 时，数据库必须是打开的，并且为当前数据库，否则 Visual FoxPro 将产生错误信息。

内部存储过程也称内部存储程序，是一段随数据库保存的代码。例如，为表设计的触发器和有效性规则都作为存储过程保存在数据库中。

【范例】打开 testdata 数据库；将如下 longdate 过程录入到文本文件 MyPro.txt 中。用 APPEND PROCEDURES 命令将 MyPro.txt 中的过程添加到 testdata 数据库中。用 DISPLAY PROCEDURES 命令显示数据库中的存储过程。最后将文本文件 MyPro.txt 删除。

MyPro.txt 中的内容为：

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

例子程序：

```
CLOSE DATABASES                          &&关闭数据库
*打开 VFP Sample 数据库 testdata
OPEN DATABASE (HOME(2) + 'Data\testdata')
MODIFY FILE MyPro.txt                    &&将 longdate 过程录入到文件 MyPro.txt 中
APPEND PROCEDURES FROM MyPro.txt         &&复制程序到数据库中
CLEAR
DISPLAY PROCEDURES                       &&显示数据库程序
DELETE FILE MyPro.txt                    &&删除中间文件 MyPro.txt
```

【相关命令和函数】COPY PROCEDURES，CREATE TRIGGER，DISPLAY PROCEDURES，MODIFY PROCEDURE，OPEN DATABASE，PROCEDURE，SET DATABASE。

AVERAGE 命令

【语法】

```
AVERAGE [ExpressionList]
[Scope] [FOR lExpression1] [WHILE lExpression2]
[TO VarList | TO ARRAY ArrayName]
```

[NOOPTIMIZE]

【用途】计算数值表达式或字段的算术平均值。

【参数描述】

ExpressionList：指定计算平均值的表达式。ExpressionList 可以是用逗号分隔当前表的字段名或包含表字段名的数值表达式。

Scope：指定要处理记录的范围。只有在指定范围内的记录才参加求平均值运算。Scope 子句可以为：ALL、NEXT nRecords、RECORD nRecordNumber 和 REST。默认的 Scope(范围)为 ALL。包含 Scope 子句的命令仅处理活动工作区中的表。

FOR lExpression1：lExpression1 为一逻辑表达式，用来作为筛选条件。只有满足逻辑条件 lExpression1 的记录才参与计算。此条件可以筛选出不想要的记录。如果 lExpression 是可优化的表达式，Rushmore 技术将优化 AVERAG FOR 查询。要获得最佳性能，应在 FOR 子句中使用可优化的表达式。

WHILE lExpression2：从第 1 个使逻辑表达式 lExpression2 的计算值为“真”(.T.)的记录开始计算，直至遇到第 1 个使表达式计算结果为“假”(.F.) 的记录为止。

TO VarList：指定保存计算值结果的变量或数组元素(下标变量)的列表。此列表中各项用逗号分隔。

TO ARRAY ArrayName：指定保存计算结果的一维数组。该数组可以在使用 AVERAGE 命令前声明。如果此处指定的数组不存在，则 Visula FoxPro 将自动创建该数组。如果数组存在，但大小不能包含所有结果，Visula FoxPro 将自动根据信息量调整数组大小。

NOOPTIMIZE：禁止本命令的 Rushmore 优化。

【使用说明】除非包含可选的表达式列表 ExpressionList，否则选定表的所有数值和货币字段都将参与平均值的运算。如果 SET TALK 为 ON，则结果将显示在屏幕上。如果 SET HEADINGS 为 ON，则在结果的上面将显示字段名或包括字段名的表达式。

【范例】打开 Visual FoxPro 例子中的 Orders.dbf 表，计算其 Freight 字段和其他数值字段的平均值。

```
CLEAR
* 打开 VFP 例子中的 Orders.dbf 表
OPEN DATABASE (HOME(2) + 'Data\testdata')
LOCAL ARRAY MyArray(2)
USE Orders                          &&打开 order 表
AVERAGE Freight                     &&计算 Freight 字段的平均值
AVERAGE  TO ARRAY MyArray           &&计算表中各数值字段的平均值
? MyArray(1),MyArray(2),MyArray(3),MyArray(4)
CLOSE DATABASES
```

【相关命令和函数】CALCULATE，DIMENSION，SET HEADINGS，SUM。

BLANK 命令

【语法】

```
BLANK
[FIELDS FieldList]
[Scope]
[FOR lExpression1]
[WHILE lExpression2]
[NOOPTIMIZE]
[IN nWorkArea | cTableAlias]
```

【用途】清除当前表中指定字段中的数据。如果执行本命令时不带任何参数，则清除当前记录中所有字段中的数据。该命令为 Visual FoxPro 7.0 增强型命令。

【参数描述】

FIELDS FieldList：FieldList 为用逗号“,”分隔的字段名列表，指出要清除数据的字段。如果省略

此为试读，需要完整PDF请访问：www.ertongbook.com