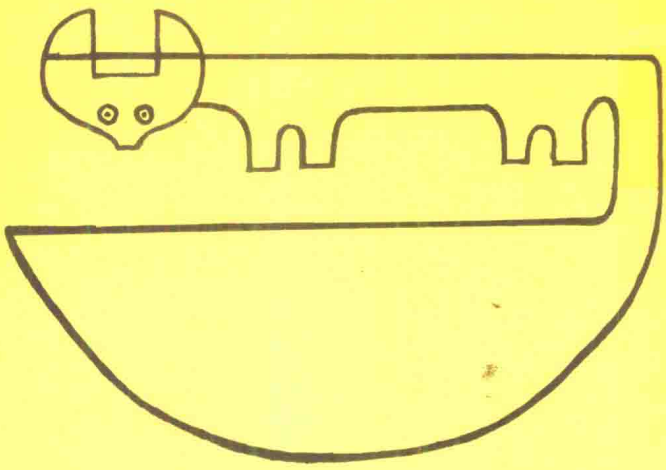
FoxPro 2.5

关系数据库语言

●李占其　高　洁　等编著

●電子工業出版社



FoxPro 2.5
关系数据库语言

李占其　高　洁　等编著

電子工業出版社

(京)新登字 055 号

内容简介

本书对 FoxPro 2.5 版的命令、函数和系统内存变量的使用方法进行了详细的叙述，并对 FoxPro 开发中的若干问题进行了讨论，同时给出了许多实例。

本书适合微机数据库系统的开发人员、FoxPro 软件用户及高校计算机专业师生参考使用。

图书在版编目(CIP)数据

FoxPro 2.5 关系数据库语言/李占其，高洁编著. -北京：
电子工业出版社，1994.8
ISBN7-5053-2408-X

Ⅰ.F…
Ⅱ.①李…②高…
Ⅲ.关系型数据库-程序语言
Ⅳ.TP312

电子工业出版社出版
北京市海淀区万寿路 173 信箱 (100036)
电子工业出版社发行　各地新华书店经销
原子能出版社激光照排中心排版
北京市顺义县天竺颖华印刷厂印刷

开本：787×1092 毫米 1/16　印张：25　字数：560 千字
1994 年 8 月第 1 版　1994 年 9 月第 1 次印刷
印数：8000 册　　定价：22.00 元

前　言

FoxPro 2.5 版是目前微机上最新的一种关系数据库管理系统，它由过去的 Foxbase 演变而来，并完全兼容 Foxbase。但是，它与 Foxbase 又有很大不同，表现在如下 3 个方面：

1. 提供了多种辅助开发工具，如菜单生成器，屏幕生成器，报表生成器等。使用这些工具，用户可在屏幕上用类似于画图的方式做出各种菜单、界面、报表，根据需要插入很少的程序片段，FoxPro 就可以自动生成菜单、界面等的源代码，使程序开发人员的编程工作量大大减少。

2. 增加了很多命令和函数。特别是提供了 SQL 命令，使 FoxPro 具有关系演算能力，这标志着 FoxPro 已是一个真正的关系数据库系统。

3. 在 Windows 环境下 FoxPro 可充分利用 Windows 的图像处理功能，使程序开发人员可以设计出十分友好、完善的应用系统。

在使用 FoxPro 2.5 版的过程中，我们对其命令、函数有了一定程度的理解，积累了一些经验体会，于是参考一些中英文资料，编成本书。简洁、明了、实用是本书的特点。希望它能对读者有较大的帮助。

本书由李占其同志负责编写，参加编写的还有高洁、邝联章、杨诚、高峰、王伟志、程重、姚丽等同志。

本书的出版得到了《计算机世界》杜海萍同志的热情帮助，特此致谢。

作　者

一九九四年三月

目　录

第一章　FoxPro 语言

§1　概　　述

FoxPro 是为数据管理而开发的一个软件产品。在系统中提供了许多辅助开发工具，使软件开发人员的工作效率能得到大幅度的提高，劳动强度降低。FoxPro 总共向用户提供了606 个命令函数，52 个系统变量，为软件开发人员创建了一个十分方便的开发环境。

在 Foxpro 系统中，用户可以使用 FoxPro 提供的各种开发工具，以非过程化的方式执行许多数据库操作及程序开发。但要充分发挥 FoxPro 的强大功能，仍需要编写相当数量的程序。另一方面，只有对 FoxPro 提供的各种命令、函数有相当程度的了解后，才能自如地使用这些工具。因为这些工具中都含有大量选项；实际上，这些选项大部分都是命令、函数中的子句或参数。

本书对 FoxPro 中的全部命令、函数和系统变量进行了简明的叙述。并把我们在实际工作中编写的程序，做为示例以帮助读者尽快进入 FoxPro 世界。

§2　命令、函数的格式约定

在 FoxPro 中，每个命令、函数都有一定的格式约定。在这些格式约定中，不同书写方式及符号都有不同的含义。

例　APPEND FROM ARRAY 〈array〉

　　[FOR 〈expL〉] [FIELDS 〈field list〉]

此命令用于从数组中向数据库追加记录。它由 3 部分组成：关键字(APPEND FROM ARRAY，FOR，FIELDS)，表达式(expL)和名(array，field list)，并且在其中用了中括号[ ]和尖括号〈 〉。

下面我们分别对关键字及各种符号进行详细叙述。

大写　大写形式的多个字符表明是 FoxPro 的关键字(保留字)，它不能当做变量来使用。例如上例中的 APPEND，FROM，FOR 等，每个命令或函数中，至少有一个保留字。

〈 〉　尖括号中的部分要用户来填写，它可以是文件名、表达式或变量等。

( )　小括号是专为函数提供的。

[ ]　中括号表示此项可有可无。例如上例中的 FOR〈expL〉。

|　竖线表示在其分割的多个选项中只能选择一项。

…　省略号表示后续部分以与前面相似的方式继续下去。

§3 表达式的组成

表达式由字段、函数、变量、常量和运算符组成。在 FoxPro 中有 4 种表达式：字符型、数字型、日期型和逻辑型，分别用〈expC〉，〈expN〉，〈expD〉，〈expL〉表示。〈expr〉是这 4 种表达式的通写形式。〈expr list〉表示一个以逗号分割的表达式表。

下面我们分别详述这 4 个表达式：

1. 字符型表达式

字符型表达式组成

- 字符型字段
- 返回字符值的函数
- 包含字符数据的内存变量和数组元素
- 字符常量

字符表达式操作符

操作符	操作
+	两个字符表达式连接
-	两个字符表达式连接(删去第一个字符串的追尾空格)
$	字符表达式比较

2. 数字表达式

数字表达式组成

- 数字类型字段
- 返回数字值的函数
- 数字型的内存变量或数组元素
- 数字常量

数字表达式操作符

操作符	操作	(按优先顺序列出)
( )	改变操作顺序	
* * | ^	幂运算	
* , /	乘、除运算	
%	取模	
+ , -	加，减运算	

3. 日期表达式

日期表达式组成

- 日期型字段
- 返回日期值的函数
- 日期型内存变量或数组元素

·日期型常量

定义一个日期型变量的方法如下：

```
STORE {//} TO LDATE1
STORE {  } TO LDATE2
```

这两条命令定义了两个空日期型变量 LDATE1,LDATE2

```
STORE {1/1/93} TO LDATE3
```

这条命令定义了一个日期型变量 LDATE3,并将其赋值为 1/1/93。

4. 逻辑表达式

逻辑表达式组成

·逻辑型字段

·返回逻辑值的函数

·逻辑型内存变量或数值元素

·用专门的关系运算符分割开的其它表达式

逻辑表达式的操作符

操作符	操作	（按优先顺序）
( )	改变运算顺序	
! |NOT	逻辑非	
AND	逻辑与	
OR	逻辑或	

逻辑表达式关系操作符

操作符	操作
<	小于
>	大于
=	等于
<>,#,! =	不等于
<=	小于等于
>=	大于等于
==	字符串恒等比较

§4 关于简写说明

下面所列出的这些简写，是在 FoxPro 命令或函数描述中通常采用的。

简写	含意
alias	别名
array	数组名
border string	边界定义串
column	屏幕或窗口列
command	FoxPro 命令

delimiter	分隔符
dir	目录
drive	磁盘驱动器名
fcode	函数代码
key	索引表达式
list	表达式、字段等的集合
maco name	键盘宏定义名称
merao field	备注字段名
memo var	内存变量
pad name	菜单笺名
parm list	参数表
path	路径
port	串/并输出端口
procedure name	过程文件名
row	屏幕或窗口行
scope	数据库记录范围
skel	文件说明框架
state ments	FoxPro 的命令集
textlines	非 FoxPro 命令的正文
topic	FoxPro 帮助文件的子题目

§5 确定记录范围的子句 scope,FOR,WHILE

scope 子句由 3 个选择组成：

- ALL——对所有的记录进行操作。
- NEXT〈expN〉——对当前记录后面〈expN〉条记录进行操作。
- RECORD〈expN〉——对数据库中第〈expN〉条记录起作用。

FOR 和 WHILE 子句的格式为 FOR 〈expL〉,WHILE 〈expL〉,对于 FOR 子句,对使逻辑表达式〈expL〉为真的记录进行操作。对于 WHILE 子句,当〈expL〉为真时才开始操作。

§6 Rushmore 技术与 FOR 子句

Rushmore 技术是一种十分高效的数据库访问技术。它可以使处理含有数百万条记录的数据库速度,与大型机相仿。而且,它使数据库记录处理的速度与被检索的记录数成正比。

在 FOR 子句出现的任何地方都可以使用 Rushmore 技术,它用于已被索引的单个数据库。在其条件表达式(为一逻辑表达式)中,该数据库字段必须是索引字段,若使用了非索引字段,则不使用 Rushmore 技术。

例 1
```
USE ASSI
INDEX ON AA1 TO PSSI
```

SUM AA3 FOR AA1='交通部' TO LL

此例中 AA1='交通部'为求合条件,因为 AA1 是索引字段,所以上例执行过程中将使用 Rushmore 技术。

例 2 USE ASSI

INDEX ON AA2 TO PSSI

SUM AA3 FOR AA1='交通部' TO LL

此例中由于 AA1 不是索引字段所以在其执行过程中不会使用 Rushmore 技术。

§7 数据库工作区

在 FoxPro 2.5 的 32 位扩充版本中,为用户提供了 255 个工作区。A～J 可以用来表示前 10 个工作区,也可以使用 1～255 数字或数据库别名来区分它们。

当在某个分区中打开了一个数据库,那么该数据库名,就是该分区的别名,除非你在打开该数据库时另外指定了其别名。

在 USE 命令中有 ALIAS 和 AGAIN 子句,用于指定打开的数据库的别名。

下面通过一个例子来说明别名的用法:

SELE C

USE ASSI

执行上述两条命令后,下面 3 条命令的执行结果是相同的,即将当前分区设置为 ASSI 所在分区。

SELE 3

SELE C

SELE ASSI

当你在当前分区中要使用其它分区中的数据库字段时,可在字段名前加上别名或分区号,中间以点号(.)或箭头号(→)分隔。例 ASSI. AA1,C→AA1。

当内存变量与数据库中字段同名时,FoxPro 总是优先考虑字段,而不是变量。

第二章　FoxPro 程序设计中的几个问题

§1　窗口和 READ 命令

每一个新 FoxPro 用户，都会对 FoxPro 中窗口的多样变化及 READ 功能的独特使用方式感到困惑。在 FoxPro 中，其界面是纯窗口式的，这是它与 Foxbase 最大的不同点之一。正是因为窗口的引入，使 FoxPro 的应用开发工作与 Foxbase 产生了很大的差异。因此，既使是熟练的 Foxbase 使用者，当他初步涉足 FoxPro 环境时，也会感到无所适从。其根本原因就是窗口问题。因此，我们先从窗口这一概念入手，并对与之密切相关的 READ 命令进行讨论，以帮助读者更好地理解 FoxPro。

所谓窗口，在 FoxPro 中就是一个有边界的区域，该区域上有标题行，在 FoxPro for Windows 中窗口左上角一般还会有一个控制箱(可以没有)，以对窗口进行控制。窗口有桌面，用户窗口，系统窗口，对话窗口和警告窗口。

在 Windows 中有多任务这一概念，它的物理含义是，在计算机上可以有多个任务放在桌面(Desk Top)上进行处理，在处理某一任务过程中，用户可以将任务暂时放下，去做另一项任务。在 FoxPro 中，对窗口的一个比较恰当的理解是：每个窗口是桌面上的一张纸，也许一个大任务要使用多个窗口，即要使用很多张纸。用户可以通过鼠标，把不同任务的作业纸调出来，执行该任务。调用方法是，用鼠标在所选窗口上点一下即可。

现在，我们就会遇到这样一个问题，READ 命令如何在多变的窗口中激活 GET 目标？在 FoxPro 中可以为每个窗口中的 GET 目标设置一个 READ，当进行 GET 处理时，激活另外一个窗口，就要引起 READ 嵌套。FoxPro 为用户提供的 READ 命令，支持多达 5 级的嵌套。

另一种 READ 用法就是多个窗口使用一个 READ 命令来激活各个窗口中的 GET 目标。此时，通过 Tab 键或箭头键可以使光标在各个窗口中的对象之间移动。

通过使用 READ 命令中 MODAL 选项或 WITH 子句可以使光标只能激活指定窗口。此时，当光标在其它窗口上进行选择时，计算机会发出响铃警告。

READ 命令中的 ACTIVATE 和 DEACTIVATE 子句，可用来根据当前窗口的变化情况执行相应的操作。DEACTIVATE 子句中的参数，用以判断是否中止当前的 READ 命令。DEACTIVATE 子句用以激活其它的窗口；ACTIVATE 子句激活新窗口中的 GET 目标。

由于 READ 命令中有许多子句，它们在执行 READ 命令时的执行顺序是不同的。

当系统中第一个 READ 命令被执行时，它的各个子句的执行顺序如下：

1. 执行 WHEN 子句，以确定是否执行 READ 操作；

2. 激活第一个含有 GET 对象的窗口；

3. 执行 ACTIVATE 子句，以激活该窗口中的 READ 命令；

4. 执行 SHOW 子句，以将目标显示出来；

5. 执行 GET 级 WHEN 子句，决定是否处理第一个 GET 对象。

在其后各个 READ 命令被执行时，它们的各个子句的执行顺序如下：

1. 执行 VALID 子句；

2. 释放 GET 字段所在窗口；

3. 激活新的 GET 字段所在窗口；

4. 执行 DEACTIVATE 子句，以决定是否终止该 READ 命令；

5. 执行 ACTIVATE 子句将新窗口中 GET 目标激活；

6. 执行 GET 级 WHEN 子句，以决定是否执行该 GET 命令。

§2 FoxPro 中颜色设置

FoxPro 的界面属窗口风格。因为窗口的引入，使界面中元素增多，例如窗口标题，窗口中的文本，窗口区域、窗口边界等，因此设置这些元素的颜色，比较麻烦。不过 FoxPro 为用户提供了非常完善的颜色设置方法。在 FoxPro 2.5 版中，直接与颜色设置有关的命令、函数有：CREATE COLOR SET，SCHEME ()，SET COLOR OF SCHEME，SET COLOR SET，SET COLOR TO，SET COLOR OF。其中 SET COLOR TO 仅为保持与 Foxbase 的兼容而保留，它可以用 SET COLOR OF SCHEME 来取代。

下面是 FoxPro 中颜色及其代码对照表。

颜　色	代　码	颜　色	代　码
黑　色	N	反向显示	I
无　色	X	洋　红	RB
蓝　色	B	红　色	R
棕　色	GR	白　色	W
青　色	GB	黄　色	GR+
绿　色	G	下划线	U

我们通过介绍四个与颜色设置有关的名词，来介绍上述与颜色设置有关的命令。

颜色对(Color Pair)

一个颜色对由两个颜色代码组成，它们之间用一斜线分割。第一个颜色代码为前景颜色，第二个颜色代码为背景颜色。例如：R/W 为一颜色对。若颜色代码中使用了星号(*)，可使颜色闪烁或将其置为高亮。在 FoxPro for Windows 中，星号(*)仅使背景色变亮，加号(+)使前景色变亮。而且还可以使用由 6 个数字组成的 RGB(红、绿、蓝)颜色值来定义一个颜色对。前 3 个数为前景色，后 3 个数为背景色。例如：RGB(255,0,0,255,255,255)等价于 R/W。

颜色对列表与颜色配置(Color Scheme)

1～10 个用逗号分割的颜色对的组合称为颜色对列表。例如 N/R，B/R，GR/B，RB/G 为一颜色对列表。一个颜色对列表分别定义了窗口中 1～10 个元素的颜色。

含有 10 个颜色对的颜色对列表称为颜色配置。它定义界面中诸如系统窗口，用户自定

义窗口，标题、菜单等 10 个元素的颜色。

在 FoxPro for MS-DOS 中，颜色配置是由 Window 菜单中 Color 选项来建立，它是一种最快的设置颜色的方法。

在 FoxPro for Windows 中，颜色配置可用控制面板中 Color 应用来设置。用户自定义的窗口，可用窗口定义中的 COLOR SCHEME 〈expN〉子句来设置颜色配置号，以实现颜色设置。注意，并不是所有的界面元素都可以用颜色配置来设置颜色，系统窗口的颜色总是由窗口控制面板的颜色设置来决定的。

颜色集(Color Set)

24 个颜色配置的集合称为颜色集。颜色集可以用 CREATE COLOR SET 命令来建立。用户可以先用 SET COLOR OF SCHEME 命令建立多个颜色设置，然后用 CREATE COLOR SET 来将这些颜色设置存入一个颜色集中。颜色集存贮在 FOX USER. DBF 源文件中。用户可以用 SET COLOR SET TO 命令来加载一个颜色集，再使用命令中的 COLOR SCHEME 子句来加载一个指定的颜色设置。可以在 CONFIG. FW 文件中加以 COLOR SET＝〈color set name〉，来使系统启动时自动加载一指定的颜色集。

§3 用户自定义函数(UDF)

UDF 是一个 FoxPro 程序，它执行一定的操作功能，并向调用程序返回一个值。在 FoxPro 2.5 版中提供了 200 多个系统函数，但是对于某些特殊的需求，还需用户编写一些自己定义的函数，以满足软件开发需要。

UDF 由函数和过程两部分内容组成。对于函数，其第一行必须是 FUNCTION 命令，标识下面的程序是一个函数，FUNCTION 后面给出的字符串是该函数的函数名。对于过程，其第一行必须是 PROCEDURE 命令，标识下面的程序是一个过程，PROCEDURE 后面给出的字符串是过程名。若 UDF 的名字与 FoxPro 函数同名，则在调用它时，系统执行 FoxPro 函数。

UDF 中使用 RETURN 命令向调用程序返回一个结果，其用法是 RETURN〈expr〉。它被省略或直接使用 RETURN 时，自动向调用程序返回真(. T.)。

在 UDF 的第二行应是 PARAMETERS 命令，用以指定一些局部变量，接收调用者发出的调用参数。变量传递有传值和传址两种方式，关于这两种方式的详细内容，可参考 PARAMETERS，PARAMETERS()，SET UDFPARMS。

许多命令中的参数，可以用 UDF 具体产生，这时，在用户的命令中就要使用 UDF。但并不是所有命令中的子句都可使用 UDF，用户在使用前可查该命令的说明。

§4 关系数据库标准语言——SQL

SQL(Structured Query Language)语言是 1974 年由 Boyce 和 Chamberlin 提出的，并在 IBM 公司 San Jose Research Laboratory 研制的 System R 上首次实现了它。它的最突出的优点是：功能丰富、使用方式灵活、语言简洁易学。1986 年 10 月，美国国家标准局(ANSI)

的数据库委员会X3H2批准了SQL作为关系数据库语言的美国标准。不久国际化标准组织(ISO)也做出了同样的决定。现在世界上绝大部分关系数据库系统都支持SQL语言。

标准的SQL语言(又称为结构化查询语言),由查询(Query)、操纵(Manipulation)、定义(Definition)和控制(Control)4部分功能组成。

在FoxPro 2.5版中提供了3条SQL命令:

查询	SELECT
定义	CREATE TABLE
操纵	INSERT

这3条命令虽然不是完整的SQL,但它已为用户提供了极大的灵活性,请用户参考本书第三章。在SQL语言中,SELECT查询语句是其核心,它功能最强,最灵活。因此,下面我们介绍标准SQL语言中的SELECT语句。

在标准SQL中,SELECT语句的一般格式是:

SELECT	目标列
FROM	基本表(或视图)
[WHERE	条件表达式]
[GROUP BY	列名1[HAVING 内部函数表达式]]
[ORDER BY	列名2{ASC / DESC}];

它的含义是:根据WHERE子句中的条件表达式,从基本表(或视图)中找出满足条件的元组,按SELECT子句中的目标列,选出元组中的分量形成结果表。如果有ORDER子句,则对查询出的结果,根据指定的列名可按升序或降序排序。GROUP子句将结果按列名1分组。下面是几种查询类型:

一、简单查询

所谓简单查询是指仅对一个基本表使用SELECT语句进行的查询。它是相对于下面连接查询和嵌套查询而言的。

二、连接查询

当查询涉及两个以上表时,则称之为连接查询。例如:

```
SELECT KK.AA1,KK.AA2,SS.LL;
  FROM KK,SS;
  WHERE KK.NO=SS.NO
```

这条命令的含义是:从基本表KK,SS中选出NO项相等的元组,然后以分量AA1,AA2,LL做为目标元组输出。WHERE后面的条件称为连接条件或连接谓词。连接谓词中的字段称为连接字段。

三、嵌套查询

嵌态查询亦称为子查询。嵌态查询是指一个SELECT-FROM-WHERE查询块中,可以嵌入另一个查询块。在SQL中允许多层嵌套。例:

```
SELECT S#,SN
```

```
FROM S
WHERE S# IN
        SELECT S#
        FROM SC
        WHERE C# IN
                SELECT C#
                FROM C
                WHERE CN= 'J'
```

上面这个例子是求选修了课程名为'J'的学生的学号和姓名。

在 FoxPro 2.5 版中的 SELECT 命令与标准 SQL 语言相比有些差异。但是标准 SQL 的 SELECT 查询功能，它都能实现。

第三章　FoxPro 命令

在 FoxPro 2.5 版中共向用户提供了 319 个命令，其中有些仅用于 FoxPro for MS-DOS，有些仅用于 FoxPro for Windows，但是绝大部分是两者通用的。这些命令使用范围的说明将在具体命令中给出。如果没有特别说明，则在 FoxPro for Windows 和 FoxPro for MS-DOS 中均可使用。

%

功能　求模操作符。

格式　〈expN1〉%〈expN2〉

参见　MOD()

注释　此操作符返回〈expN1〉除以〈expN2〉后所得余数。它与＋，－，＊，/，^ 一样，都是算术运算符，但它的优先级与＊，/相同。〈expN1〉为被除数，它的小数位数决定了结果中的小数位数。〈expN2〉为除数，它的正负决定了结果的正负。

例
```
? (-5)%(-2)
-1
? 6.0%4
2
```

&

功能　宏替换。

格式　&〈memvar〉[.〈expC〉]

注释　宏替换把一个字符型变量的内容，做为字符串来处理。它可用名表达式来代替。

例
```
STORE 'ASSI' TO LL1
STORE 'AA1' TO LL2
USE & LL1 ORDER & LL2
```

或

```
STORE 'ASSI' TO LL1
STORE 'AA1' TO LL2
USE (LL1) ORDER (LL2)
```

上面两例运行结果相同。

〈memvar〉用来存放替换字符串。它不能递归地调用自己，而且在循环过程中，仅在循环的开始被计算一次，在以后的重复过程中，即使变量发生了变化也不会影响循环。

〈.expC〉它用来给一个宏添加额外字符〈expC〉。〈expC〉本身也是一个宏。

$

功能　判断一字符串是否是另一字符串的子串。

格式　〈expC1〉$〈expC2〉

返回　逻辑型

参见　AT()

注释　若〈expC1〉是〈expC2〉的子串，则返回真(.T.)，否则返回假(.F.)。〈expC1〉，〈expC2〉可以是备注字段。$不使用 Rushmore 技术。

例　EDIT FOR '交通部' $ AA4

其中 AA4 是一备注字段。

=

功能　计算一个或多个表达式的值。

格式　=〈expr1〉[,〈expr2〉…]

参见　?,UDF

注释　命令=计算一个或多个表达式〈expr1〉,〈expr2〉…〈exprN〉的值。在一个函数〈不管是 FoxPro 函数还是用户定义函数〉需要某种效果，但同时不必将函数的返回值送入一个内存变量或字段的情况下，这一命令是非常有用的。

例　把 3 号文件关闭，并把缓冲区中的数据写到磁盘上，可用下面的命令：

=FCLOSE(3)

INSMODE 通常返回一个真(.T.)或假(.F.)值。在此例中，该函数被执行，但却抛弃了其返回值。

等号=在逻辑表达式中用作比较运算，它还可以用于对数组或内存变量赋值。在这两种情况中，并不把符号作为一条命令看待。

参见　?,用户定义函数。

\/\\

功能　输出一行文本。

格式　\〈文本行〉|\\〈文本行〉

参见　-PRETEXT,SET TEXTMERGE,-TEXT,SET TEXTMERGE DELIMITERS,TEXT…ENDTEXT

注释　此命令用于将一行文本输出到屏幕和正文文件中。此命令可用于建立一个文本文件。使用 SET TEXTMERGE TO 命令，可以指定要建立的文本文件的名字。如果 SET TEXT MERGE 设置为 OFF，则输入只送到屏幕上。使用带 NOSHOW 选项的 SET TEXTMERGE，也可以取消文本行在屏幕上的显示。\与\\的区别在于前者输出的文件行带有回车换行符，后者则不带。

在文本行中可以使用表达式，但它必须包括在《》符号之中，而且 SET TEXTMERGE 必须被设置为 ON，这时才对表达式取值。

例　SET TEXTMERGE TO LL.TXT