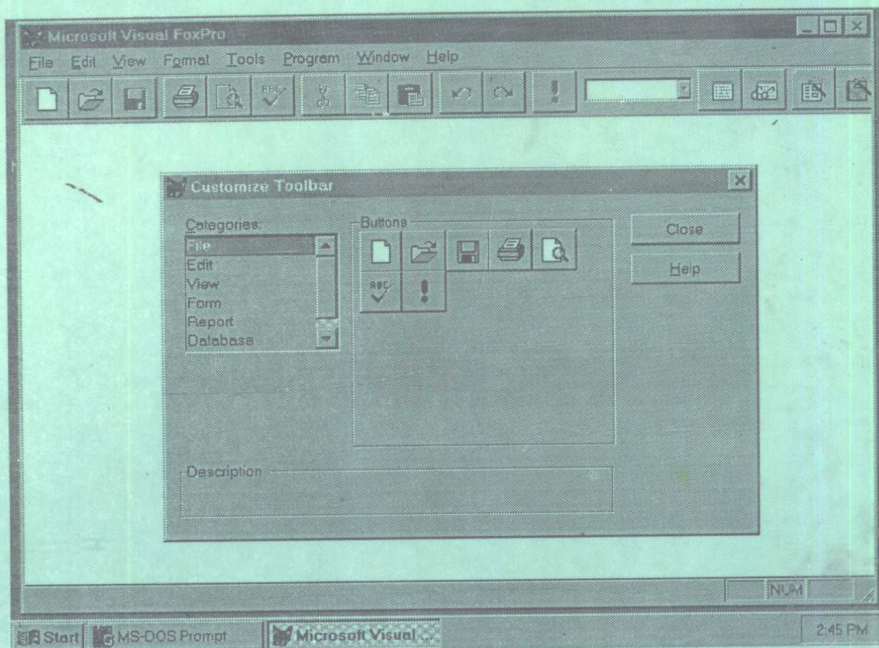


运通创作室编著



Visual FoxPro 3

语言实用详解

—— 命令篇

希望

学苑出版社

Visual FoxPro 3.0

语言实用详解

——命令篇

运通创作室	编著
吴 言	审校

学苑出版社

内 容 提 要

Visual FoxPro 3.0 是 FoxPro 的新一代产品,具有快速开发应用程序、面向对象和客户机/服务器等强大功能。为了方便读者在使用和编程时参考,我们特别推出了“Visual FoxPro 3.0 语言实用详解”系列书,本书是该套书中的一本,主要介绍 Visual FoxPro 3.0 的命令、操作符和预处理指令的参考信息,按字母顺序解释它们的作用、语法、参数、使用的方法和相关的项目,还给出了一些示例。该书是用 Visual FoxPro 3.0 开发应用程序的读者必备的工具书。

需要本书的读者,可与北京海淀 8721 信箱书刊部联系,电话 2562329, 邮政编码 100080。

Visual FoxPro 3.0 语言实用详解

——命令篇

编 著:运通创作室
审 校:吴 言
责任编辑:甄国宪
排 版:运通创作室
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:双青印刷厂
开 本:787×1092 1/16
印 张:60.625 字 数:1374 千字
印 数:1~5000 册
版 次:1995 年 11 月北京第 1 版第 1 次
ISBN7-5077-1049-1/TP·48
本册定价:85.40 元(套/三册)

学苑版图书印、装错误可随时退换

前 言

Microsoft Visual FoxPro 3.0 是一个 32 位的数据库开发系统,运行于 Windows 3.1(具有 Win32s)、Windows 95 和 Windows NT 操作系统。与 FoxPro 2.5/2.6 相比,它是一个革命性的软件产品,引进了可视编程和面向对象的概念。

Visual FoxPro 可借助工具条、对象、可视控件来自动完成界面的设计并执行各种任务,同时不牺牲任何数据库性能。不再只通过代码来完成任务,新环境让读者图形化地直接操作文件、表、对象和类。在可视环境下工作,可以通过鼠标拖动来编写程序。

不仅如此,Visual FoxPro 还可重复使用各种类,直观地、创造性地建立应用程序。可以从工具条上访问 OLE 控件和定制控件,从而可以利用 Microsoft Office 和其它应用程序的各种功能。可以在 Visual FoxPro 应用程序中修改和运行 Microsoft Excel 电子表格。

另外,可视化的工具和加强的连接性,使用户可以与大多数后台数据库的客户机/服务器应用程序连接,使 32 位的 ODBC 驱动程序能够集成来自各个系统的数据,包括客户机/服务器数据库的数据、本地数据以及其它应用程序的数据。内建的工具可使本地数据升级到 Microsoft SQL Server。

Visual FoxPro 3.0 与 FoxPro 2.5/2.6 完全兼容,在 2.5 和 2.6 下面开发的应用程序不需修改就可以移植到 Visual FoxPro 3.0 中来。

为了方便读者在编程和使用时参考,我们对有关 Visual FoxPro 3.0 的大量资料进行了分析、消化和整理,结合自己的使用经验,特别推出这一套“Microsoft Visual FoxPro 3.0 语言实用详解”系列书,包括“类和对象篇”、“命令篇”和“函数和全程变量篇”三本。《类和对象篇》主要介绍 Visual FoxPro 新增的面向对象功能的参考,解释类、对象、控件、特性、方法和事件的功能和用法;《命令篇》解释 Visual FoxPro 命令、操作符、指令与菜单命令的功能与用法;而《函数和全程变量篇》则主要讨论 Visual FoxPro 的函数和全程变量的功能与用法。三本书组合在一起是一套完整的 Visual FoxPro 语言参考书。

由于时间仓促,书中错误在所难免,恳请读者在使用过程中批评指正。

运通创作室

Visual FoxPro 性能总览

Visual FoxPro 是一个功能强大的数据库管理系统(DBMS),它同以前的数据库管理系统相比,具有更快速、更有效、更灵活的突出特点。它能够迅速而又简单地建立用户的数据库,从而方便地使用和管理数据;它不仅支持客户机/服务器(C/S)结构,而且具有高度与其他软件(如 Excel、Word)共享和交换数据的能力。Visual FoxPro 提供新的对象和事件处理模式,利用面向对象编程(OOP)的威力使用户能够最快速地建立和修改应用程序。最后,Visual FoxPro 对以前版本的 FoxPro 提供完全的兼容性,用户以前的应用程序可完全不经修改直接在 Visual FoxPro 上运行。

Visual FoxPro 的优点

简单、易学、易用

1. 快速完成应用任务

Visual FoxPro 提供了向导(Wizard)、生成器(Builder)和设计器(Designer)三种工具,这三种工具都使用图形交互界面方式,使用户能够最简单而又最快地完成数据操作任务。

操作向导(Wizard)提供了用户要完成某项工作所需的详细操作步骤,在这些步骤的指导下,用户可以一步步地很简单地完成任务。例如,用户可用表向导(Table Wizard)来帮助建立一个数据表,用窗体向导(Form Wizard)来建立窗体,而查询向导(Query Wizard)将指示用户建立一个标准查询所需的完整步骤。

生成器(Builder)也是一种具有友好界面的图形工具,它的主要功能是在用户自己的应用程序中加入一定的控制功能。例如列表框生成器(ListBox Builder)就是一个带有标签(tab)的对话框界面,利用列表框生成器,用户可以在窗体中设计出一个列表框,并且可以在这种生成器中设置一个列表框的共同特性。

如果用户想突破向导和生成器本身的限制,想要自己对应用程序进行更复杂或更灵活的控制,可以利用另一种 Visual FoxPro 提供的方便有效的工具——设计器(Designer)。设计器也提供了一个友好的图形应用程序开发接口,通过它用户能建立起自己的应用程序。例如,用户可以用窗体设计器(Form Designer)定义和生成一个窗体,用数据表设计器(Table Designer)定义和生成一个数据表。

2. 使用方便的工具栏

像许多其他 Microsoft 产品一样,Visual FoxPro 也给用户提供了使用方便的工具栏(Toolbars),工具栏里有许多按钮,它们代表着菜单里的某些选项。一般来说,用户经常执行的操作(如 Open file)或使用的对象(如 Command Windows)都对应一个按钮,用户可以通过选择这些按钮方便而迅速地完成任务,而不必通过菜单选项。

另外,用户可以自己定制 Visual FoxPro 中的工具栏,增加或减少一些按钮,还可以在自己建立的应用程序中定义和实现方便用户使用的工具栏。

3. 不编程而建立应用程序界面

Visual FoxPro 提供的窗体设计器(Form Designer)是一种功能强大的工具,用户能够不编程或使用很少的代码来实现友好的交互式应用程序界面以及对界面的控制。例如用户可以用栅格控件(Grid control)很容易地建立一对多的窗体;用户只需把一个数据表拖动到一个窗体上就可以了。也可以利用页格式控件(PageFrame control)来建立有标签的对话框或用户自己的生成器界面。

4. 用项目管理器统一管理工作

Visual FoxPro 提供的另一高效易用的工具是项目管理器(Project Manager),通过项目管理器,用户可以集中地管理数据、文档、类库、源代码等各种资源。例如用户可以建立和更新数据库,设计或改变窗体和报表,定义或改变类库,生成或重新生成自己的应用程序。另外,用户也能在项目管理器中使用 Visual FoxPro 提供的简单而有效的其他工具,如向导、生成器、工具栏等。所有这些,使用户能够对各种工作进行集中管理而又简单有效。

功能更强大

Visual FoxPro 比以前的数据库管理系统具有更强大的功能。它通过使用快速查询(Rushmore)技术和对系统的优化而使用户最大限度地体会到 Visual FoxPro 快速而又功能强大的优点。

1. 具有面向对象编程的能力

Visual FoxPro 在支持标准的 Xbase 传统编程方式的同时,也提供了完全的面向对象编程(OOP)能力。在 Visual FoxPro 的对象模式下,用户可以利用所有的面向对象编程特性,这些特性包括继承(inheritance)、封装(encapsulation)、多态性(polymorphism)以及分类(sub-classing),它们都作为用户所熟悉的 Xbase 编程语言的扩展集而实现。

Visual FoxPro 提供了一些基类(class),包括窗体、工具栏、页格式等,使用这些类,用户可以建立基本的窗体、工具栏或页格式,这样就可以一方面减少用户编程工作量,另一方面又加快程序开发过程。

再进一步,用户可以将自己定义的类再进行分类,这样可利用用户已有的源代码或窗体。例如,用户可以将基本的窗体类再进行分类而建立自己的子类,这个子类将根据用户的要求自动地在应用程序中建立起一个用户希望看到的窗体,它的结构是由用户分类决定的。

Visual FoxPro 类模式能够在用户应用程序中对对象进行深入而全面的控制,例如用户在设计时可用窗体设计器对窗体中的对象进行完全的控制,而类模式下当用户运行程序时可对窗体中对象的表现和行为提供相同的控制。

在 Visual FoxPro 中,用户可以用类设计器(Class Designer)交互式地建立一个类,或者用 DEFINE CLASS 命令来编程建立。

2. 更容易处理事件

Visual FoxPro 包含一种事件模式,它能够帮助用户自动地处理事件。在这种事件模式下,用户可以获取并控制所有标准的 Windows 事件,例如鼠标的移动。通过处理这一事件,用户可以拖动和放置一个对象。用户可以用两种方法来控制事件:一种是通过特性窗口(Properties Window)来可视地控制;另一种是通过 Visual FoxPro 的编程语言来控制。这两种方法都能使用户很容易地建立起完全的事件驱动应用程序而不用考虑 READ 层次及浏

览窗口限制,也不用编写事件处理程序。

3. 最优化系统

Visual FoxPro 能够通过优化用户的系统设计来提高自身的性能。在所有的优化措施中,最有效的方法是尽可能多地增加用户的扩展内存(extended memory)或者减少被其他应用程序(如 Windows)所占用的内存。其余的提高 Visual FoxPro 性能的措施包括加快启动速度和优化设置(SET)命令。

4. 使用快速查询技术

快速查询(Rushmore)技术是一种专用的数据查询技术,它能够迅速地从数据库中选择出一组满足用户要求的记录。使用这种技术能将数据查询所需的时间从几小时或几分钟减少到几秒钟,这样可以极大地提高数据查询的效率。

支持客户机/服务器结构

Visual FoxPro 可作为开发强大的客户机/服务器(C/S)应用程序的前台。Visual FoxPro 既支持高层次的对服务器数据的浏览,又提供了对本地服务器语法的直接访问,这种直接访问给用户提供了开发灵活的客户机/服务器应用程序的坚实的基础。Visual FoxPro 提供了支持客户机/服务器结构所需的各种特性:多功能的数据词典、本地和远程视图、NULL 值支持、事务处理、对任何 ODBC 数据资源的访问。

1. 用数据词典定义规则

Visual FoxPro 数据库(.DBC 文件)提供了一个数据词典,使用这个数据词典,用户可以数据库中的每一个数据表添加规则、视图、触发器、永久关系和连接。

在一个数据库中,用户可以定义:

- (1) 字段级或记录级的规则,这种规则将在用户的应用程序使用该数据表时起作用。
- (2) 主索引键和候补索引键。
- (3) 本地和远程视图。
- (4) 触发器。
- (5) 数据表之间的永久关系。
- (6) 对远程数据资源的连接。
- (7) 存储进程。
- (8) 字段的缺省值。
- (9) 长表名及字段名。

另外,用户可以通过引用完整(Referential Integrity)生成器来定义插入、更新和删除规则,这样可以加强每一个保存关系的引用完整性。

Visual FoxPro 也支持数据表中的 NULL 值,这种能力极大地提高了 Visual FoxPro 同其他数据资源的兼容性和连接能力,这些数据资源包括 Microsoft Access, Visual Basic 及基于 SQL 的服务器。

2. 查看远程或异种数据

用户可以使用来自远程、本地或多数据表的异种数据,以便在用户的本地计算机上开发和测试一个客户机/服务器应用程序。本地数据视图使用本地计算机上的数据表而不是远程服务器上的数据表。而多表数据查看使用的是多个不同数据表中的相关数据。为了减少用

户从服务器上卸载的数据量,用户可以建立带参数的视图,然后从用户的 Visual FoxPro 客户机/服务器应用程序中更新远程数据。

3. 用事务处理来控制共享访问

共享访问是指多个用户对数据的共享以及相应的一些必要的访问限制,例如为了不让某用户访问某些数据,用户可以建立起支持数据共享访问的应用程序。用户在建立应用程序时,如果使用事务处理和缓冲手段(记录级或数据表级),则可以减少编程的工作量。Visual FoxPro 内含的批处理进程和详细的对更新冲突处理的控制可以使多用户环境中的数据更新过程得以简化。

4. 实现客户机/服务器应用程序

在客户机/服务器应用程序开发中,用户除了使用数据视图之外,还可以通过 Visual FoxPro 的 SQL 通路功能来发送当前服务器所识别的控制台命令,这样用户可以直接访问服务器。这种功能比数据视图提供了更多的对服务器的访问和控制。

Visual FoxPro 具有将用户的应用程序升档的能力。升档是指用户在本地机上建立一个应用程序后,可以基于一个后台的数据资源使应用程序运行在一个客户机/服务器环境中,这样做的好处之一就是用户可以用和本地的 Visual FoxPro 数据表结构一样的结构建立起远程的服务器数据库。不仅如此,用户在升档时可以选择哪些数据表放在服务器中而哪些表放在本地机上,这样可以既提供共享能力,又提高访问效率。

同其他软件的高度兼容性

Visual FoxPro 可以同其他 Microsoft 软件共享数据,例如用户可用自动 OLE 来包含其他软件(如 Excel、Word)中的对象并在 Visual FoxPro 中使用这些软件。

1. 同其他软件共享数据

在 Visual FoxPro 中同其他软件共享数据是很容易的。用户可用主元表向导(PivotTable Wizard)使 Excel 共享 Visual FoxPro 数据,还可用邮件合并向导(MailMerge Wizard)使 Word 共享 Visual FoxPro 数据。

用户可以通过在表或窗体中联接其他软件中的对象来直接对这些对象进行编辑而不用退出 Visual FoxPro 环境。

2. 输入和输出数据

用户能够在 Visual FoxPro 和其他软件之间输入和输出数据,输入数据是指 Visual FoxPro 利用其他软件生成的数据,输出数据是指 Visual FoxPro 生成一定的数据以供其他软件使用。这种输入输出是通过不同的文件格式的转换来实现的,不同的文件格式包括文本、电子表格(spreadsheets)和数据表。在 Visual FoxPro 中,用户可用输入向导(Import Wizard)来帮助决定使用哪一种文件格式。

3. 使用自动 OLE 控制其他软件

Visual FoxPro 提供的自动 OLE 能够加强用户应用程序的功能。用户可以通过编程来运行其他的软件。例如用户可以调用 Excel 来完成某些计算,命令 Graph 将运行结果绘制成图,然后把图存放在一个 Visual FoxPro 表的通用字段中,所有这些工作都可通过 Visual FoxPro 的编程来实现。

Visual FoxPro 的向下兼容性

Visual FoxPro 具有与以前的 FoxPro 版本的完全的兼容性,用户不用担心以前在 FoxPro 中开发应用程序的投资是否会浪费。在 Visual FoxPro 中,用户可以一点不修改地运行 FoxPro 应用程序。因为 Visual FoxPro 语言的扩展集(OOP 编程等)并不影响向下兼容性,所以用户也可以用 Visual FoxPro 语言来改写以前的 FoxPro 应用程序,而且可以将 FoxPro 下的屏幕、项目和报表转换为 Visual FoxPro 格式。

虽然 Visual FoxPro 具有良好的兼容性,但如果用户想利用一些 Visual FoxPro 所特有的优点的话,建议用户还是将自己的 FoxPro 文件转换为新的 Visual FoxPro 格式。

目 录

前言	I
Visual FoxPro 性能总览	II
#DEFINE... #UNDEF 预处理器指令	1
#IF... #ENDIF 预处理器指令	1
#INCLUDE 预处理器指令	3
\$ 操作符	3
% 操作符	4
& 命令	4
&& 命令	5
* 命令	5
= 命令	6
? ?? 命令	6
??? 命令	9
@...BOX 命令	9
@...CLASS 命令	10
@...CLEAR 命令	11
@...EDIT —— 编辑框命令	11
@...FILL 命令	17
@...GET —— 复选框命令	17
@...GET —— 组合框命令	26
@...GET —— 命令按钮命令	35
@...GET —— 列表框命令	45
@...GET —— 单选按钮命令	52
@...GET —— 旋转器命令	60
@...GET —— 文本框命令	65
@...GET —— 透明按钮命令	71
@...MENU 命令	77
@...PROMPT 命令	77
@...SAY 命令	78
@...SAY —— 图片和 OLE 对象命令	82
@...TO 命令	84
\\命令 ACCEPT 命令	87
ACTIVATE MENU 命令	88
ACTIVATE POPUP 命令	89
ACTIVE SCREEN 命令	90
ACTIVE WINDOW 命令	90
ADD CLASS 命令	92
ADD TABLE 命令	93
APPEND 命令	93
APPEND FROM 命令	94
APPEND FROM ARRAY 命令	97
APPEND GENERAL 命令	99
APPEND MEMO 命令	100
APPEND PROCEDURES 命令	101
ASSIST 命令	101
AVERAGE 命令	102
BEGIN TRANSACTION 命令	103
BLANK 命令	103
BROWSE 命令	104
BUILD APP 命令	116
BUILD EXE 命令	117
BUILD PROJECT 命令	118
CALCULATE 命令	119
CALL 命令	121
CANCEL 命令	122
CD CHDIR 命令	122
CHANGE 命令	123
CLEAR 命令	124
CLOSE 命令	125
CLOSE MEMO 命令	127
COMPILE 命令	127
COMPILE DATABASE 命令	128
COMPILE FORM 命令	129
CONTINUE 命令	129
COPY FILE 命令	130
COPY INDEXES 命令	130
COPY MEMO 命令	131
COPY PROCEDURES 命令	131
COPY STRUCTURE 命令	132
COPY STRUCTURE EXTENDED 命令	132
COPY TAG 命令	133
COPY TO 命令	134
COPY TO ARRAY 命令	137
COUNT 命令	138
CREATE 命令	139
CREATE CLASS 命令	140

CREATE CLASSLIB 命令	140	DISPLAY MEMORY 命令	201
CREATE COLOR SET 命令	141	DISPLAY OBJECTS 命令	202
CREATE CONNECTION 命令	141	DISPLAY PROCEDURES 命令	203
CREATE CURSOR-SQL 命令	142	DISPLAY STATUS 命令	203
CREATE DATABASE 命令	144	DISPLAY STRUCTURE 命令	205
CREATE FORM 命令	144	DISPLAY TABLES 命令	206
CREATE FROM 命令	146	DISPLAY VIEWS 命令	207
CREATE LABEL 命令	146	DO 命令	207
CREATE MENU 命令	147	DO CASE...ENDCASE 命令	209
CREATE PROJECT 命令	149	DO FORM 命令	210
CREATE QUERY 命令	150	DO WHILE...ENDDO 命令	211
CREATE REPORT 命令	151	EDIT 命令	211
CREATE REPORT-Quick Report 命令	152	EJECT 命令	222
CREATE SCREEN 命令	153	EJECT PAGE 命令	222
CREATE SCREEN-Quick Form 命令	153	END TRANSACTION 命令	223
CREATE SQL VIEW 命令	154	ERASE 命令	223
CREATE TABLE-SQL 命令	155	ERROR 命令	224
CREATE TRIGGER 命令	158	EXIT 命令	225
CREATE VIEW 命令	159	EXPORT 命令	225
DEACTIVATE MENU 命令	160	EXTERNAL 命令	227
DEACTIVATE POPUP 命令	161	FILER 命令	230
DEACTIVATE WINDOW 命令	161	FIND 命令	230
DECLARE 命令	162	FLUSH 命令	231
DECLARE-DLL 命令	162	FOR...ENDFOR 命令	231
DEFINE BAR 命令	164	FUNCTION 命令	232
DEFINE BOX 命令	168	GATHER 命令	233
DEFINE CLASS 命令	169	GETEXPR 命令	234
DEFINE MENU 命令	172	GO/GOTO 命令	235
DEFINE PAD 命令	175	HELP 命令	236
DEFINE POPUP 命令	179	HIDE MENU 命令	237
DEFINE WINDOW 命令	184	HIDE POPUP 命令	237
DELETE 命令	191	HIDE WINDOW 命令	238
DELETE CONNECTION 命令	192	IF...ENDIF 命令	239
DELETE DATABASE 命令	192	IMPORT 命令	240
DELETE FILE 命令	192	INDEX 命令	241
DELETE-SQL 命令	193	INPUT 命令	244
DELETE TAG 命令	194	INSERT 命令	245
DELETE TRIGGER 命令	194	INSERT _SQL 命令	245
DELETE VIEW 命令	195	JOIN 命令	246
DIMENSION 命令	195	KEYBOARD 命令	247
DIR 或 DIRECTORY 命令	196	LABEL 命令	248
DISPLAY 命令	198	LIST 命令	249
DISPLAY CONNECTIONS 命令	199	LIST CONNECTIONS 命令	250
DISPLAY DATABASE 命令	200	LIST DATABASE 命令	251
DISPLAY FILES 命令	200	LIST OBJECTS 命令	252

LIST PROCEDURES 命令	252	ON PAGE 命令	291
LIST TABLES 命令	253	ON READERROR 命令	291
LIST VIEWS 命令	254	ON SELECTION BAR 命令	292
LOAD 命令	254	ON SELECTION MENU 命令	292
LOCAL 命令	256	ON SELECTION PAD 命令	293
LOCATE 命令	256	ON SELECTION POPUP 命令	293
LPARAMETERS 命令	257	ON SHUTDOWN 命令	294
MD MKDIR 命令	258	OPEN DATABASE 命令	294
MENU 命令	258	PACK 命令	295
MENU TO 命令	259	PACK DATABASE 命令	296
MODIFY CLASS 命令	260	PARAMETERS 命令	296
MODIFY COMMAND 命令	261	PLAY MACRO 命令	297
MODIFY CONNECTION 命令	263	POP KEY 命令	297
MODIFY DATABASE 命令	263	POP MENU 命令	298
MODIFY FILE 命令	264	POP POPUP 命令	298
MODIFY FORM 命令	266	PRINT JOB...ENDPRINT JOB 命令	299
MODIFY GENERAL 命令	267	PRIVATE 命令	299
MODIFY LABEL 命令	268	PROCEDURE 命令	300
MODIFY MEMO 命令	269	PUBLIC 命令	301
MODIFY MENU 命令	271	PUSH KEY 命令	301
MODIFY PROCEDURE 命令	272	PUSH MENU 命令	302
MODIFY PROJECT 命令	273	PUSH POPUP 命令	302
MODIFY QUERY 命令	274	QUIT 命令	302
MODIFY REPORT 命令	276	READ 命令	303
MODIFY SCREEN 命令	277	READ MENU 命令	306
MODIFY STRUCTURE 命令	278	RECALL 命令	306
MODIFY VIEW 命令	278	REGIONAL 命令	307
MODIFY WINDOW 命令	279	REINDEX 命令	308
MOUSE 命令	280	RELEASE 命令	308
MOVE POPUP 命令	281	RELEASE BAR 命令	309
MOVE WINDOW 命令	281	RELEASE CLASSLIB 命令	309
NOTE 命令	282	RELEASE LIBRARY 命令	309
ON APLABOUT 命令	282	RELEASE MENUS 命令	310
ON BAR 命令	283	RELEASE MODULE 命令	310
ON ERROR 命令	283	RELEASE PAD 命令	311
ON ESCAPE 命令	284	RELEASE POPUPS 命令	311
ON EXIT BAR 命令	284	RELEASE PROCEDURE 命令	312
ON EXIT MENU 命令	285	RELEASE WINDOWS 命令	312
ON EXIT PAD 命令	285	REMOVE CLASS 命令	313
ON EXIT POPUP 命令	285	REMOVE TABLE 命令	313
ON KEY 命令	286	RENAME 命令	314
ON KEY= 命令	286	RENAME CLASS 命令	314
ON KEY LABEL 命令	288	RENAME CONNECTION 命令	314
ON MACHelp 命令	290	RENAME TABLE 命令	315
ON PAD 命令	290	RENAME VIEW 命令	315

REPLACE 命令	316	SET COLOR OF 命令	351
PEPLACE FROM ARRAY 命令	317	SET COLOR OF SCHEME 命令	351
RESTORE FROM 命令	318	SET COLOR SET 命令	352
RESTORE MACROS 命令	318	SET COLOR TO 命名	353
RESTORE SCREEN 命令	319	SET COMPATIBLE 命令	353
RESTORE WINDOW 命令	319	SET CONFIRM 命令	354
RESUME 命令	320	SET CONSOLE 命令	355
RETRY 命令	320	SET CPDIALOG 命令	355
RETURN 命令	321	SET CURRENCY 命令	356
RESTORE FROM 命令	321	SET CURSOR 命令	356
RESTORE MACROS 命令	322	SET DATABASE 命令	356
RESTORE SCREEN 命令	322	SET DATASESSION 命令	357
RESTORE WINDOW 命令	323	SET DATE 命令	357
RESUME 命令	323	SET DEBUG 命令	358
RETRY 命令	324	SET DECIMALS 命令	358
RETURN 命令	324	SET DEFAULT 命令	358
ROLLBACK 命令	325	SET DELETED 命令	359
RUN 命令	325	SET DELIMITERS 命令	360
RUNSCRIPT 命令	327	SET DEVELOPMENT 命令	360
SAVE MACROS 命令	328	SET DEVICE 命令	361
SAVE SCREEN 命令	328	SET DISPLAY 命令	361
SAVE TO 命令	329	SET DOHISTORY 命令	362
SAVE WINDOWS 命令	329	SET ECHO 命令	363
SCAN...ENDSCAN 命令	330	SET ESCAPE 命令	363
SCATTER 命令	331	SET EXACT 命令	364
SCROLL 命令	332	SET EXCLUSIVE 命令	365
SEEK 命令	333	SET FDOW 命令	365
SELECT 命令	334	SET FIELDS 命令	366
SELECT-SQL 命令	334	SET FILTER 命令	367
SET 命令	342	SET FIXED 命令	368
SET ALTERNATE 命令	342	SET FORMAT 命令	368
SET ANSI 命令	343	SET FULLPATH 命令	368
SET APLABOUT 命令	343	SET FUNCTION 命令	369
SET AUTOSAVE 命令	344	SET FWEK 命令	369
SET BELL 命令	344	SET HEADINGS 命令	370
SET BLINK 命令	345	SET HELP 命令	370
SET BLOCKSIZE 命令	345	SET HELPFILTER 命令	371
SET BORDER 命令	346	SET HOURS 命令	373
SET BRSTATUS 命令	347	SET INDEX 命令	373
SET CARRY 命令	347	SET INTENSITY 命令	375
SET CENTURY 命令	348	SET KEY 命令	375
SET CLASSLIB 命令	348	SET KEYCOMP 命令	376
SET CLEAR 命令	349	SET LIBRARY TO 命令	379
SET CLOCK 命令	349	SET LOCK 命令	380
SET COLATE 命令	350	SET LOGERRORS 命令	380

SET MACDESKTOP 命令	381	SET SYSMENU 命令	408
SET MACHELP 命令	382	SET TALK 命令	410
SET MACKEY 命令	382	SET TEXTMERGE 命令	410
SET MARGIN 命令	383	SET TEXTMERGE DELIMITERS 命令	412
SET MARK OF 命令	383	SET TOPIC 命令	413
SET MARK TO 命令	385	SET TOPIC ID 命令	413
SET MEMOWIDTH 命令	385	SET TRBETWEEN 命令	413
SET MESSAGE 命令	385	SET TYPEAHEAD 命令	414
SET MOUSE 命令	386	SET UDFPARMS 命令	414
SET MULTILOCKS 命令	387	SET UNIQUE 命令	415
SET NEAR 命令	387	SET VIEW 命令	415
SET NOCPTRANS 命令	388	SET VOLUME 命令	416
SET NOTIFY 命令	388	SET WINDOW OF MEMO 命令	416
SET NULL 命令	389	SET XCMDFILE 命令	417
SET ODOMETER 命令	389	SHOW GET 命令	417
SET OPTIMIZE 命令	389	SHOW GETS 命令	419
SET ORDER 命令	390	SHOW MENU 命令	420
SET PALETTE 命令	392	SHOW OBJECT 命令	421
SET PATH 命令	392	SHOW POPUP 命令	422
SET PDSETUP 命令	393	SHOW WINDOW 命令	422
SET POINT 命令	394	SIZE POPUP 命令	424
SET PRINTER 命令	395	SKIP 命令	425
SET PROCEDURE 命令	397	SORT 命令	425
SET READBORDER 命令	398	SQL 命令概览	428
SET REFRESH 命令	398	STORE 命令	428
SET RELATION 命令	399	SUM 命令	429
SET RELATION OFF 命令	400	SUSPEND 命令	430
SET REPROCESS 命令	400	System Menu Names	430
SET RESOURCE 命令	401	TEXT...ENDTEXT 命令	441
SET SAFETY 命令	402	TOTAL 命令	442
SET SCOREBOARD 命令	402	TYPE 命令	443
SET SECONDS 命令	403	UNLOCK 命令	444
SET SEPARATOR 命令	403	UPDATE 命令	445
SET SHADOWS 命令	403	UPDATE-SQL 命令	445
SET SKIP 命令	404	USE 命令	446
SET SKIP OF 命令	404	VALIDATE DATABASE 命令	449
SET SPACE 命令	406	WAIT 命令	450
SET STATUS 命令	406	WITH...ENDWITH 命令	451
SET STATUS BAR 命令	407	ZAP 命令	452
SET STEP 命令	408	ZOOM WINDOW 命令	452
SET STICKY 命令	408		

#DEFINE... #UNDEF 预处理器指令

创建和释放编译时的常量。

语法

```
#DEFINE ConstantName eExpression
```

```
...
```

```
#UNDEF ConstantName
```

参数

ConstantName

指定编译时的一个常量名。该常量名必须是一个以字母或下划线开头的最多由 254 个字母、数字或下划线组成的合法的 Visual FoxPro 名。为了提高程序的可读性并简化调试,用户的常量名使用标准的命名约定。

常量名不要使用 Visual FoxPro 的关键字。若要终止由 #DEFINE 创建的文本常量替换,使用 #UNDEF ConstantName。

eExpression

指定编译时的常量值。eExpression 可以是一个名字或一个其值为字符、数字、流通货币、日期、时间或逻辑量的表达式。

eExpression 不要使用系统内存变量。系统内存变量直到运行时才赋值。

说明

#DEFINE 和 #UNDEF 预处理器指令在程序中创建编译时的常量。用 #DEFINE 创建常量来代替使用内存变量可以减少内存消耗,提高效率并简化程序。

用 #DEFINE 创建一个常量,用 ConstantName 指定常量名,用 eExpression 指定其值。当程序编译时,就执行文本替换,即无论常量名出现在程序何处都由常量表达式代替。可通过使用 #UNDEF 来停止该常量替换。

替换只发生在 #DEFINE 语句创建该常量之后,并在该常量的 #UNDEF 语句之前的程序中。常量只对创建该常量的程序有效。

#IF... #ENDIF 预处理器指令

编译时有条件地包含源代码。

语法

```
#IF nExpression1 | lExpression1 Commands
```

```
[ #ELIF nExpression2 | #ELIF lExpression2 Commands
```

```
...
```

```
#ELIF nExpressionN | #ELIF lExpressionN Commands]
```

```
[ #ELSE
```

```
Commands]
```

```
#ENDIF
```

参数

IF nExpression1 | lExpression1 Commands

nExpression1 指定求值的数字表达式。

- 如果表达式值非零,则紧跟在 # IF 后的命令包含进编译代码。退出 # IF... # ENDIF 结构, # ENDIF 后的第一行程序被编译。
- 如果表达式值为零,则紧跟在 # IF 后的命令不被包含进编译代码。在这种情况下,执行后面的任何指令。

lExpression1 指定待求值的逻辑表达式。

- 如果表达式值为真(. T.),则紧跟在 # IF 后的命令包含进编译代码。退出 # IF... # ENDIF 结构, # ENDIF 后的第一行程序被编译。
- 如果表达式值为假(. F.),则紧跟在 # IF 后的命令不被包含进编译代码。在这种情况下,执行 # ELIF 后面的任意指令。

不要为 nExpression1 或 lExpression1 指定系统内存变量。系统内存变量直到运行时才赋值。

ELIF nExpression2 | # ELIF lExpression2 Commands

...

ELIF nExpressionN | # ELIF lExpressionN Commands

如果 nExpression1 是 0 或 lExpression1 是假(. F.),则执行 # ELIF 指令,第一个 # ELIF 表达式 nExpression2 或 lExpression2(如果有的话)被求值。如果 nExpression2 非零或 lExpression2 为真(. T.),则紧跟 # ELIF 后的命令被包含进编译代码。退出 # IF... # ENDIF 结构, # ENDIF 后的第一行程序被编译。

如果 nExpression2 为 0 或 lExpression2 为假(. F.),则 # ELIF 后的命令不包含进编译代码,执行下一个 # ELIF 指令。

ELSE Commands

如果没有 # ELIF 指令或虽有但其值为 0 或假(. F.),则有无 # ELSE 决定了是否有附加的命令包含进编译代码:

- 如果有 # ELSE,则 # ELSE 后的命令就包含进编译代码。
- 如果没有 # ELSE,则 # IF 和 # ENDIF 间没有任何命令包含进编译代码。退出 # IF... # ENDIF 结构,编译从 # ENDIF 后的第一行程序继续。

ENDIF

表示 # IF 语句的结束。

说明

IF... # ENDIF 可以提高源代码的可读性,减小编译程序大小,并在某些情况下提高效率。

当编译 # IF... # ENDIF 结构时,该结构内的后继逻辑或数字表达式被求值。求值结果决定了在编译代码中包含哪一组 Visual FoxPro 命令。

#INCLUDE 预处理器指令

告诉 Visual FoxPro 预处理器将指定的头文件的内容当作出现在 Visual FoxPro 程序中。

语法

```
#INCLUDE cFileName
```

参数

cFileName

指定编译时嵌入该程序的头文件名。将文件名用引号引起来。

可以在头文件名中带路径。当在头文件名中带路径时, Visual FoxPro 只在指定的位置搜寻该头文件。

如果头文件名不带路径, 则 Visual FoxPro 在缺省的 Visual FoxPro 目录中搜寻该头文件, 然后在 Visual FoxPro 路径中搜寻。Visual FoxPro 路径由 SET PATH 指定。

说明

用户可以创建一个头文件包含预处理器指令, 然后在程序编译时用 #INCLUDE 将头文件内容嵌入该程序。当编译到程序中出现 #INCLUDE 的地方时, 头文件的内容加入到程序中。

在头文件中只识别 #DEFINE... #UNDEF 和 #IF... #ENDIF 预处理器指令。头文件中的备注和 Visual FoxPro 命令被忽略。

一个程序可以包含任意数目的 #INCLUDE 指令, 并且这些指令可以出现在程序的任何地方。#INCLUDE 指令也可以出现在头文件中, 允许用户嵌套 #INCLUDE 指令。

典型的头文件有 .H 扩展名, 虽然它们可以有任何扩展名。Visual FoxPro 头文件 FOX-PRO.H 被包含进去, 它含有贯穿本手册讲述的许多常量。

\$ 操作符

如果一个字符表达式包含在另外一个字符表达式中, 则返回真 (.T.)。

语法

```
cSearchFor $ cSearchIn
```

返回值

逻辑值。

参数

cSearchFor

指定在 cSearchIn 中寻找的表达式。

cSearchIn

指定搜寻是否其中含有 cSearchFor 的表达式。

如果在 cSearchIn 中找到 cSearchFor, 则返回真 (.T.); 否则, 返回假 (.F.)。cSearchFor 和 cSearchIn 可为字符类型的内存变量或数组元素、字符类型的字段、字符串文字或任意长