

Visual Foxpro 3 Developer's Guide

VISUAL FOXPRO 3

开发指南

(美) Jeb Long 著
京京翻译组 译

计算机软件开发
与程序设计
系列丛书



机械工业出版社



西蒙与舒斯特国际出版公司

34
1
计算机软件开发与程序设计系列丛书

Visual FoxPro 3 开发指南

(美) Jeb Long 著

京京翻译组 译

机械工业出版社
西蒙与舒斯特国际出版公司

本书讲述了怎样利用 Visual Foxpro 3 提供的面向对象的编程环境开发出通用的和功能强大的交互式界面,还介绍了 Visual Foxpro 3 为表单选择大量可视化对象,和所提供的数据库字典、触发器、无缝的客户/服务器数据库连接,以及许多功能强大的向导。

可供软件开发人员、计算机用户参考和使用。

Jeb Long: Visual FoxPro 3——Developer's Guide/3rd Edition

Authorized translation from the English language edition published by SAMS.

Copyright 1996 by SAMS.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签,无标签者不得销售。

版权所有,翻印必究。

本书版权登记号 图字 01-96-1221

图书在版编目(CIP)数据

Visual FoxPro 3 开发指南 / 美 朗 (Long J.) 著 京京翻译组译 —北京 机械工业出版社, 1997. 1

(计算机软件开发与程序设计系列丛书)

ISBN 7-111-05465-2

I. V... II. ①朗... ②京... III. 关系数据库-数据库管理系统, FoxPro 3-软件开发-指南 IV. TP311. 13-62

中国版本图书馆 CIP 数据核字 (96) 第 23781 号

出 版 人: 马九荣 (北京市百万庄南街 号 邮政编码 00037)

责任编辑: 何伟新 高科

三河永和印刷有限公司印刷 · 新华书店发行

1997 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16 · 67.5 印张 · 1685 千字

0 001-5000 册

定价: 119.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

自从美国微软公司(Microsoft)于1993年6月发布FoxPro 2.5 for Windows之后,各种个人计算机上的Xbase系统开始从DOS平台向Windows平台转移。FoxPro 2.5 for Windows是对Xbase一个出色的延伸,在FoxPro 2.5下,以MS-DOS为基础的dBASE III PLUS和dBASE IV 2.0程序几乎不作任何改动便可直接运行。

1994年3月,Microsoft公司又推出了与dBASE IV 2.0版几乎完全兼容的FoxPro 2.6。在FoxPro 2.6中,提供了一系列“向导”(Wizard),它们可以帮助用户简便地完成工作。

1995年春天,Visual FoxPro问世了,它提供了真正的非模态界面。不仅能把多个表单简便地合成到一起,而且还能同时运行一个表单的多个实例。Visual FoxPro可以帮助开发者对事件进行控制,这样就能用一个面向对象的编程环境开发出通用的和功能强大的交互式界面,这种界面比用以前版本的FoxPro开发出来的任何一种界面都要好。除此以外,可以为表单选择大量可视化对象,比如工具栏、网格、嵌套网格以及三维按钮等等。Visual FoxPro也提供了一个数据字典,它可以强制关系的完整性;触发器,对它进行编程便可实现对交叉表完整性进行控制;无缝的客户/服务器数据库连接;以及许多功能强大的向导。

为了帮助我国计算机用户熟练地使用Visual FoxPro,为提高我国计算机的应用水平,我们翻译了这本《Visual FoxPro 3 开发指南》。由于时间仓促,翻译过程中难免出现错误,欢迎广大读者指正。

京京翻译组

1996.10 北京

序

欢迎 Jeb Long 加入 FoxPro 作者的行列。Jeb 先生是 Xbase 的倡导者之一。另外,作为 JPLDIS(dBASE II 的前身)的作者,他发明了许多重要的命令,比如 REPLACE、STORE 等等,这些命令直到如今仍在使用。我很高兴地看到他现在开始使用 FoxPro 了。

第一次与 Jeb 结识的时候,正值 Ashton-Tate 公司尝试推出 Fox 软件,那次尝试后来流产了,当时是 1986 到 1987 年间。在那段时间里,我与 Jeb 的关系逐渐地密切,他在软件工程学上的造诣给我留下了深刻的印象。从那时以后,我们就成了非常要好的朋友。

Jeb 的新书《Visual FoxPro 3 开发指南》第三版把重点放在 FoxPro 语言上面,这个策略是相当正确的。尽管 Jeb 透彻讨论了由 FoxPro 提供的强有力的工具,但他的大多数时间仍然花在强调语言的运用上面——通过程序化的编程或者通过代码片断进行阐述。这与最近出版的、与 FoxPro 有关的大多数书籍是一种有趣的对比。显然,Jeb 希望为读者清晰地展示 FoxPro 语言的各种组件相互之间是怎样联系起来的。

祝贺 Jeb 写作成功这本《Visual FoxPro 3 开发指南》第三版。对于任何一名 FoxPro 开发者来说,收藏这本参考书都是相当有价值的。

David L. Fulton

注:1. David L. Fulton 为 FoxPro 原创者。——译者

作者简介

Jeb Long 先生在美国某些著名的机构里面有着 30 年的软件开发和工程设计经验。自从 1990 年离开 Ashton-Tate 公司以后,Long 先生就以一名独立撰稿人和咨询顾问的身份工作着。就职于 Ashton-Tate 公司的时候,Jeb Long 从事 dBASE 产品的相关指导工作,他是 dBASE 语言的设计者之一。Long 先生还是 dBASE III 和 dBASE IV 所有版本(除最初的版本以外)的 dBASE 语言组件的负责人。

Long 先生与 dBASE 的关系可以追溯到 1973 年,那时,他是加利福尼亚工程技术学院喷射推动实验室的一名软件工程师。他的任务之一就是为 JPL 的 UNIVAC 1108 大型计算机开发一个文件管理程序。这个程序就是 JPLDIS,即 dBASE 的前身。后来,根据与 Wayne Ratliff 签订的协议,Long 先生翻译和改编了 dBASE II 的原始版本,使其能在 IBM PC 机上运行。

Long 先生在喷射推动实验室工作了 11 年,他为美国行星探索程序的软件支持作出了巨大的贡献。在他完成的各种软件项目中,涉及到用于计算行星轨迹的程序、处理太空飞行器遥测数据的程序、太空船推动子系统仿真程序以及太空船的测试和分析程序等等。他还积极参与了对火星进行探索的 Mariner and Viking 科研项目。

Long 先生于 1963 年毕业于得克萨斯州州立大学,目前与夫人 Elizabeth 居住于能够俯瞰洛杉矶的一座小山上。Elizabeth 已经获得了英语硕士学位,她现在是 JPL(喷射推动实验室)的一名专业作家。写作本书的时候,Jeb 和 Elizabeth 是“亲密的战友”。Long 先生还编著了许多其他书籍,并为技术杂志撰写了数量可观的文章。他的书包括《自学 Quick C for Windows》,Sams Publishing 出版社;《自学 Microsoft C/C++ 7》,Sams Publishing 出版社;《dBASE IV 1.5 程序语言》(与 Alastair Dallas 合作编著),Sams Publishing 出版社;以及《dBASE 编程工具》(与 R. A. Byers 和 C. Wayne Ratliff 合作编著),Ashton-Tate Publishing 出版社。另外,Sams Publishing 还出版了由 Jeb Long 编写的四本 FoxPro 开发指导书,这些书包括《FoxPro 2.5 for Windows 开发指南》、《FoxPro 2.5 for DOS 开发指南》、《FoxPro 2.6 for Windows 开发指南》以及本书。

2/5/08/op

前 言

欢迎阅读《Visual FoxPro 3 开发指南》第三版,这本书可以帮助您利用 Visual FoxPro 使用和开发应用程序。无论对于初级 FoxPro 用户还是有经验的 FoxPro 用户,本书都是很有帮助的,它能帮助大家掌握用 Visual FoxPro 开发应用程序所需的技能。假如读者已经掌握了 Microsoft Windows、编程以及数据库的一些常识,阅读本书就能达到最佳的效果。

Visual FoxPro 是运行于 IBM PC 计算机系统内的一种计算机程序,它必须在 Microsoft Windows 环境下才能正常地发挥作用。运行 Visual FoxPro 的计算机的基本配置是:

- 一个 80386SX 或者更好的处理器
- 一个鼠标
- 640KB 常规内存
- 8MB RAM
- Microsoft Windows 3.1 或更高版本
- 约 56MB 硬盘空间。根据不同的安装选项,对磁盘空间的要求也会发生变化

Visual FoxPro 是一种数据库管理系统(DBMS),利用它可以对一个数据列表进行建立和维护,我们把这个数据库列表叫作“表格”(table),或者直接称呼为“数据库文件”。这种数据列表的例子包括电话号码清单、存货清单或者食品清单等等。事实上,我们可以用 FoxPro 对相关信息的任何一个集合进行维护;我们可以对表格进行排序或者组织;可以在一张表格内搜索并取回所需的信息;可以通过表格生成报表、输入表单以及邮件标签等等。我们能对包含了常规信息的表格进行关联,FoxPro 可以自动对这种关系进行维护,以便以后根据需要处理或者取回有关的信息。根据一种不严格的定义,我们把这种类型的 DBMS 叫作“关系型 DBMS”;而一组相关的表格组合起来就构成了一个“数据库”。

FoxPro 使用了一些易于掌握的命令来实现数据的建立和处理以及从数据库表格内获取数据。FoxPro 隶属于一种名为 Xbase 的 DBMS 类别。Xbase 这个术语是很常见的,它表示诸如 FoxPro、dBASE III PLUS、dBASE IV、FoxBASE+以及 ARAGO 的语言。Xbase 最初起源于大型机使用的一个 DBMS,这个 DBMS 叫作 JPLDIS(喷射推动实验室数据库管理和信息检索系统)。该 DBMS 由本书的作者于 1972 年开发成功。到目前为止,全世界的开发者和程序员已经编写了几千万行的 Xbase 代码。它已经成为许多事务处理在程序语言上的首要选择。根据 Borland International 公司的报告,全世界约有四百万人拥有 dBASE IV 的拷贝——dBASE IV 是当今流行的 Xbase 语言之一。在 Visual FoxPro 3.0 下,dBASE IV 程序不进行任何改动便可直接运行。但是,这种程序在 FoxPro 下的运行速度会更快。自从微软收购了 Fox Holdings 公司以后,FoxBASE+和 FoxPro 这两个产品得到了长足的发展,FoxPro 在 Xbase 中占据的市场份额得到了显著的提升。Visual FoxPro 支持由其他 Xbase 产品的命令;另外,它还对语言进行了充分的扩展,这种扩展是其他任何一种 Xbase 产品尚未提供的。在 Visual

FoxPro 对 Xbase 语言作出的所有扩展里,最基本的就是面向对象程序设计(OOP)以及支持数据字典的编程元素。

Visual FoxPro for Windows 的问世是微型计算机发展历史的一个里程碑,因为现在不进行任何改动,用 80 年代的 MS-DOS 事务语言——Xbase——编写的程序就可以直接运行于 90 年代的操作环境——Windows 3.1 或者 Windows 95 图形用户界面(GUI)下,而且可以在一个客户/服务器环境中运行。Visual FoxPro 支持无缝连接。除此以外,我们还可以充分利用新的、功能强大的面向对象程序设计环境,从而改进应用程序的开发方式。

本书的读者

本书适用于所有 FoxPro 用户,其中包括 FoxPro 的初学者到有经验的专业人士,后者曾经使用过以前版本的 FoxPro。Visual FoxPro 3.0 是一种新产品,这本书将就它的特性进行介绍。由于对每一种特性都进行了详细的讨论,所以无论是 FoxPro 的初学者还是 FoxPro 专家,都能从本书受益。其他某些书籍把讨论的重点放在怎样利用 FoxPro 语言实际问题上面,但这种方式并不利于全面掌握 FoxPro 语言,而这本书的目的是帮助大家更好地理解 Visual FoxPro 3.0 语言。语言掌握得越好,理解得越透彻,Visual FoxPro 对自己就会越有用。

许多 FoxPro 参考书是为那些连计算机的基本操作都还没有熟悉的新手准备的,本书并非这种类型。尽管在这本书里对 Visual FoxPro 的所有方面都进行了介绍,但是讨论的焦点主要放在 Visual FoxPro 的界面、编程语言以及开发工具上面,并且提供了在其他地方找不到的一些深层次的信息。初学者熟悉了计算机的概念,并且接受了入门手册的指导以后,就可以接着阅读本书。这本书可以帮助他们对 Visual FoxPro 语言、用户界面以及开发工具有一个更加深入的理解。

这本书也适用于另外一种类型的读者——希望快速掌握 Visual FoxPro 语言的程序员,他们没有必要先去阅读入门手册。本书可以帮助大家抓住 Visual FoxPro 语言的本质。

FoxPro for Windows 的发展

FoxPro 2.5 for Windows 于 1993 年 6 月发布。这是 Xbase 一个重要的里程碑,因为它是在 Microsoft Windows 平台中运行的第一个真正的 Xbase 系统。由于这个版本的推出,许多 PC 开发者都开始转向 Microsoft Windows 平台。以后的情况是显而易见的——MS-DOS 市场很快就走上了下坡路。1993 到 1994 年间,Borland International 公司在 dBASE IV(在 MS-DOS 下运行)上的收入下降了 40%还要多。从 1995 年开始,Borland 公司放弃了以 MS-DOS 为基础的任何 dBASE 版本的开发计划。

FoxPro 2.5 for Windows 是对 Xbase 一个出色的延伸,在 FoxPro 2.5 下,以 MS-DOS 为基础的 dBASE III PLUS 和 dBASE IV 2.0 程序几乎不作任何改动便可直接运行。然而,在大多数情况下,为了使以前的 dBASE 应用程序能在 FoxPro 2.5 for Windows 下正常运行,必须对它们进行改动。

FoxPro 2.6 是 1994 年 3 月推出的,该软件与 dBASE IV 版本 2.0 几乎是完全兼容的。即使有不兼容的地方,也显得不是很重要。除此以外,FoxPro 2.6 提供了一系列“向导”(Wizard),它们可以帮助用户简便地完成工作。在这儿,“向导”是一种进程,它能引导用户完成一些典型的数据库操作。FoxPro 2.6 for Windows 包含了大量“向导”,利用它们可以建立表

格和大量报表、标签、邮件合并报表、屏幕以及客户/服务器查询和本地查询。

在 FoxPro for Windows 2.6 下,利用标准的 Windows 界面控件元素和对话框,我们可以开发出非常不错的应用程序。然而,这种应用程序是有限制的,因为 FoxPro 2.6 仍然沿用了与 FoxPro 的 MS-DOS 版本类型相同的伪事件驱动可视化界面。把多个表单合并起来仍然是相当困难的。

1995 年春天,Visual FoxPro 问世了,它解决了以前 FoxPro 版本带来的问题,提供了真正的非模态界面。我们不仅能把多个表单简便地合成到一起,而且还能同时运行一个表单的多个实例。Visual FoxPro 可以帮助开发者对事件进行控制,这样一来,我们就能用一个面向对象的编程环境开发出通用的和功能强大的交互式界面,这种界面比用以前版本的 FoxPro 开发出来的任何一种界面都要好。除此以外,针对自己的表单,我们可以为它选择大量可视化对象,比如工具栏、网格、嵌套网格以及三维按钮等等。Visual FoxPro 也提供了一个数据字典,它可以强制关系的完整性;提供了触发器,对它进行编程便可实现对交叉的表完整性进行控制;提供了无缝的客户/服务器数据库连接;并且提供了许多功能强大的向导。

Visual FoxPro 3.0 for Windows 的新特性

Visual FoxPro 3.0 进行了大量重要的改进,并提供了许多新特性。所有现成的设计器(Designer)以及构建器(Builder)的功能都得到了增强。在 Visual FoxPro 3.0 里,有一种名为数据库容器(Database Container,DBC)的新文件类型,其中包含的信息可以支持关系的完整性、触发器以及存储进程,以便把表格与数据库关联到一起。它也支持用于表格的扩展属性,比如数据有效性校验、长字段名以及缺省值等等。Visual FoxPro 3.0 的主要革新在于它在编程方法上的重新定位。Visual FoxPro 抛弃了传统的过程式程序设计,改为采用面向对象的程序设计。针对熟悉 FoxPro 早期版本的人士,笔者提供了一份清单,其中列出了 Visual FoxPro 3.0 语言具备的几乎所有新特性。其中新增的大多数新命令和函数都用于支持面向对象程序设计(OOP)和数据集成:

类名	说明
AClass()	AClass()函数的作用是用对象的父类名填写一个内存变量数组。
ADatabase()	ADatabase()函数的作用是用所有已打开数据库的名字以及它们的路径填写一个内存变量数组。
ADOBJECTS()	ADOBJECTS()函数的作用是用一个数据库里面已命名的所有连接、表格或者 SQL 视图的名字填写一个内存变量数组。
ADD CLASS	ADD CLASS 函数的作用是在 VCX 可视类库里面附加一个指定的类定义。
ADD TABLE	ADD TABLE 命令可以在当前打开的数据库内增添一个自由表格。
AERROR()	AERROR()函数的作用是建立一个内存变量数组,其中包含的信息与 Visual FoxPro 或者 ODBC 最近出现的错误有关。
AINSTANCE()	AINSTANCE()函数可以用某个指定类的所有实例名字来填写一个内存变量数组。
ALTER TABLE	ALTER TABLE 命令可以程序化地修改一个表格的结构。
AMEMBERS()	AMEMBERS()函数的作用是用某个对象的属性、进程以及成员对象

APPEND PROCEDURES	来填写一个内存变量数组。 APPEND PROCEDURES 命令可以把一个正文文件内的进程添加到当前打开数据库的存储进程里面。
APRINTERS()	APRINTERS()函数的作用是用 Windows 打印管理器内当前安装的打印机的名字填写一个内存变量数组。
ASELOBJ()	ASELOBJ()函数的作用是在一个内存变量数组里面填写对活动 Form Designer 内当前选中控件的一个引用。
AUSED()	AUSED()函数可以在一个内存变量数组里面填写一个对话区的表格别名和工作区。
BEGIN TRANSACTION	BEGIN TRANSACTION 命令的作用是启动一次事务处理。
BIT...()	这里增加了九个函数,它们用于针对数值执行二进制位运算(一次处理一个二进制位)。这些函数包括 BITAND(), BITCLEAR(), BITLSHIFT(), BITNOT(), BITOR(), BITRSHIFT(), BITSET(), BITTEST()以及 BITXOR()。
_BUILDER	_BUILDER 系统内存变量包含了一个串,其中存储了 Visual FoxPro 构建器应用程序的名字。
CANDIDATE()	假如某个索引标志是一个候选索引标志,CANDIDATE()函数就会返回一个真值(.T.)。
CD CHDIR	保留下来的 MS-DOS 命令,CD 和 CHDIR 用于把缺省目录改变成自己希望的目录。
CLEAR EVENTS	CLEAR EVENTS 命令可以中断正在处理的事件,这个事件最初是由 READ EVENTS 命令激活的。
CLOSE TABLES	CLOSE TABLES 命令的作用是关闭当前数据库内的所有表格,同时保持数据库的打开状态不变。
COMPILE DATABASE	COMPILE DATABASE 命令可以对一个数据库容器(DBC)文件内的存储进程进行编译。
COMPILE FORM	COMPILE FORM 命令的作用是对 Visual FoxPro 表单(SCX)文件内的一个或者多个表单进行编译。
COMPOBJ()	COMPOBJ()函数可以对两个对象的属性进行比较,假如它们的属性和属性值完全一致,就会返回真值(.T.)。
_CONVERTER	_CONVERTER 系统内存变量包含了 FoxPro 转换器应用程序的名字。
COPY PROCEDURES	COPY PROCEDURES 命令的作用是把当前数据库内的存储进程拷贝到一个正文文件里面。
CREATE CLASS	CREATE CLASS 命令的作用是打开 Class Designer,在其中可以自行建立一个类定义。
CREATE CLASSLIB	CREATE CLASSLIB 命令可以建立一个新的、空的可视类库文件。
CREATE CONNECTION	CREATE CONNECTION 命令的作用是建立一个命名连接,并把它存储到当前的数据库内。
CREATE DATABASE	CREATE DATABASE 命令的作用是建立一个数据库,并且打开它。
CREATE FORM	CREATE FORM 命令的作用是打开新的 Form Designer,用它可以生成一个屏幕(SCX)文件。与 MODIFY SCREEN 命令的功能相同。
CREATE SQL VIEW	CREATE SQL VIEW 命令的作用是显示 View Designer,从中可以建

CREATE TRIGGER	立一个 SQL 视图。 CREATE TRIGGER 命令可以针对某张表格建立一个删除、插入或者更新触发器。
CREATEOBJECT()	CREATEOBJECT()函数可以根据一个类或者子类定义建立一个对象,也可以建立一个 OLE 对象。
CTOT()	CTOT()函数可以根据一个字符表达式返回 DateTime(日期/时间)值。新的 DateTime 数据类型同时包含了日期和时间信息,比如:{12/12/95 10:34:04 am}。
CURSORGETPROP()	CURSORGETPROP()函数的作用是返回一张 FoxPro 表格或者游标当前的属性。
CURSORSETPROP()	利用 CURSORSETPROP()函数可以设置 FoxPro 表格或者游标的属性。
CURVAL()	CURVAL()函数的作用是直接从磁盘或者一个远程资源处返回一个字段值。
DATETIME()	DATETIME()函数可以将当前的日期和时间信息当作一个 DateTime 值返回。
DBC()	DBC()函数可以返回当前数据库的名字和路径。
DBGETPROP()	DBGETPROP()函数的作用是返回当前数据库或者当前数据库内字段、命名连接、表格或者视图的一种属性。
DBSETPROP()	利用 DBSETPROP()函数,我们可以设置当前数据库或者当前数据库内的字段、命名连接、表格或者视图的一种属性。
DBUSED()	假如指定的数据库(DBC 文件)处于打开状态,DBUSED()函数便会返回一个真值(.T.)。
DECLARE - DLL	DECLARE - DLL 命令的作用是注册一个函数,这个函数位于某个外部 Windows 32 位动态链接库(DLL)内。
DEFINE CLASS	DEFINE CLASS 命令的作用是建立一个用户自定义类或者子类,并为这个类或者子类指定属性、事件和方法。
DELETE CONNECTION	DELETE CONNECTION 命令的作用是从当前数据库内删除一个命名连接。
DELETE DATABASE	DELETE DATABASE 命令的作用是将一个数据库从磁盘上删除掉。
DELETE - SQL	DELETE - SQL 命令的作用是把记录标记成删除状态。
DELETE TRIGGER	DELETE TRIGGER 命令的作用是从当前数据库内删除用于表格的一个 Delete、Insert 或者 Update 触发器。
DELETE VIEW	DELETE VIEW 的作用是从当前数据库内删除一个 SQL 视图。
DISPLAY CONNECTIONS	DISPLAY CONNECTIONS 命令的作用是显示出与当前数据库内命名连接有关的信息。
DISPLAY DATABASE	DISPLAY DATABASE 命令的作用是显示出与当前数据库或者当前数据库内的字段、命名连接、表格或者视图有关的信息。
DISPLAY OBJECTS	DISPLAY OBJECTS 命令的作用是显示出一个对象或者一组对象有关的信息。
DISPLAY PROCEDURES	DISPLAY PROCEDURES 命令的作用是显示当前数据库内存储进程的名字。
DISPLAY TABLES	DISPLAY TABLES 命令可以显示出一个打开数据库内的所有表格,

DISPLAY VIEWS	以及这些表格有关的信息。 DISPLAY VIEWS 命令的作用是显示与当前数据库内 SQL 视图有关的信息。
DO FORM	DO FORM 命令的作用是执行一个编译表单或者通过 Form Designer 建立的表单集。
DTOT()	DTOT() 函数可以通过一个日期表达式返回相应的 DateTime 值。
END TRANSACTION	END TRANSACTION 命令的作用是中止当前的事务处理。
ERROR	ERROR 命令的作用是生成一个指定的 FoxPro 出错。
FDATE()	FDATE() 函数的作用是返回某个文件最近修改的日期。
FTIME()	FTIME() 函数的作用是返回某个文件最近修改的时间。
GETCOLOR()	GETCOLOR() 函数可以显示出 Windows Color 对话框,并将选中的颜色返回成一个数值。
GETCP()	GETCP() 函数的作用是显示 Code Page (代码页) 对话框,并返回选中代码页的编号。
GETFLDSTATE()	GETFLDSTATE() 函数可以返回一个数值,用它指出表格内的一个字段或者游标是否已被编辑,或者当前记录的删除状态是否已发生了改变。
GETNEXTMODIFIED()	GETNEXTMODIFIED() 函数的作用是返回某个记录的编号,这是在缓冲游标内准备修改的下一个记录。
GETOBJECT()	GETOBJECT() 函数的作用是激活一个 OLE 自动对象,并建立对那个对象的一个引用。
GETPRINTER()	GETPRINTER() 函数的作用是激活 Windows Print Setup 对话框,并且返回所选打印机的名字。
HOUR()	HOUR() 函数的作用是从 DateTime 表达式里面返回对应的小时部分。
#INCLUDE	#INCLUDE 命令的作用是指示 Visual FoxPro 预处理程序对指定文件的内容进行特殊对待——使这些内容好象在 FoxPro 程序里面显示一样。
ISEXCLUSIVE()	假如某个打开的表格具有排它性用途,ISEXCLUSIVE() 函数就会返回一个真值(.T.);否则返回假值(.F.)。
INDBC()	假如目标数据库对象位于当前的数据库内,INDCB() 函数就会返回一个真值(.T.);否则返回假值(.F.)。
ISMOUSE()	假如系统内已安装了鼠标,ISMOUSE() 函数就会返回一个真值(.T.);否则返回的就是一个假值(.F.)。
ISNULL()	假如某表达式被赋予一个空(NULL)值,ISNULL() 函数就会返回一个真值(.T.);否则它返回的就是一个假值(.F.)。
LIST CONNECTIONS	LISTCONNECTIONS 命令的作用是连续性地显示出与当前数据库内命名连接有关的信息。
LIST DATABASE	LIST DATABASE 命令的作用是显示出与当前数据库有关的信息。
LIST OBJECTS	LIST OBJECTS 命令的作用是显示出与一个或者一组对象有关的信息。
LIST PROCEDURES	LIST PROCEDURES 命令的作用是显示出当前数据库内存储进程的名字。
LIST TABLES	LIST TABLES 命令的作用是显示出所有表格以及与一个打开数据库

LIST VIEWS	LIST VIEWS 命令的作用是显示出与当前数据库内 SQL 视图有关的信息。
LOCAL	利用 LOCAL 命令可以建立局部内存变量和内存变量数组。
LPARAMETERS	LPARAMETERS 命令的作用是把从一个调用程序内传递过来的数据分配给局部内存变量或者数组。
MD MKDIR	保留下来的 MS-DOS 命令,MD 和 MKDIR 命令可用于建立指定的磁盘目录。
MESSAGEBOX()	MESSAGEBOX()函数的作用是显示一个用户自定义对话框。
MINUTE()	MINUTE()函数的作用是从 DateTime 表达式里面返回对应的分钟部分。
MODIFY CLASS	MODIFY CLASS 命令将显示出 Class Designer,用户在其中可对一个现成类定义进行修改,或者建立一个新的类定义。
MODIFY CONNECTION	MODIFY CONNECTION 命令将显示出 Connection Designer,用户在其中可对一个现成的命名连接进行修改,这个连接存储于当前的数据库里面。
MODIFY DATABASE	MODIFY DATABASE 命令将打开 Database Designer,这样便可对当前的数据库进行交互式修改。
MODIFY FORM	MODIFY FORM 命令的作用是打开 Form Designer,以便用户修改或者建立一个表单。
MODIFY PROCEDURE	MODIFY PROCEDURE 命令的作用是打开 Visual FoxPro 正文编辑器,这样便可针对当前数据库建立新的存储进程,或者对当前数据库内的现成存储进程进行修改。
MODIFY VIEW	MODIFY VIEW 命令的作用是显示出 View Designer,这样便可对一个现成的 SQL 视图进行修改。
MOUSE	MOUSE 命令用于对鼠标进行单击、双击、移动或者拖动操作。该命令在生成演示和教程的时候非常有用。
MTON()	MTON()函数的作用是从一个货币表达式里面返回一个数值。
NTOM()	NTOM()函数的作用是从一个数值表达式里面返回一个货币值。
NVL()	NVL()函数可以通过两个表达式返回一个非空值。
OBJREF()	OBJREF()函数的作用是用一个内存变量建立对某个对象的非直接引用,或者建立一个数组元素,使其用作对某个对象的引用。
OLDVAL()	OLDVAL()函数的作用是返回一个字段使用的最初的字段值,那个字段已被修改,但是未作更新。
OPEN DATABASE	OPEN DATABASE 命令的作用是打开一个数据库。
PACK DATABASE	PACK DATABASE 命令的作用是从一个打开的数据库内实际删除那些已做上删除标记的记录。
PRIMARY()	假如某个索引标记是主索引标记,PRIMARY()函数就会返回一个真值(.T.);否则返回的就是一个假值(.F.)。
RD RMDIR	RD 和 RMDIR 命令可用于磁盘上指定的目录。
READ EVENTS	READ EVENTS 命令的作用是启动事件处理。
REFRESH()	REFRESH()函数可用于对某个远程的、可更新的 SQL 视图内的记录集合进行刷新。

RELEASE CLASSLIB	RELEASE CLASSLIB 命令的作用是关闭包含了类定义的一个 VCX 可视类库。
RELEASE PROCEDURE	RELEASE PROCEDURE 命令的作用是关闭由 SET PROCEDURE 命令打开的一个进程文件。
REMOVE CLASS	REMOVE CLASS 命令的作用是从 VCX 可视类库里面删除一个类定义。
REMOVE TABLE	REMOVE TABLE 命令的作用是从当前的数据库里面删除一张表格。
RENAME CLASS	RENAME CLASS 命令的作用是对 VCX 可视类库里面包含的某个类定义进行更名处理。
RENAME CONNECTION	RENAME CONNECTION 命令的作用是对某个数据库里面的一个命名连接进行更名处理。
RENAME TABLE	RENAME TABLE 命令的作用是对某个数据库内的一张表格进行更名处理。
RENAME VIEW	RENAME VIEW 命令的作用是对数据库内的一个 SQL 视图进行更名处理。
REQUERY()	REQUERY() 函数可以重复从某个远程 SQL 视图里面获取数据的操作。
RGB()	RGB() 函数的作用是从一系列红、绿、蓝颜色元素里面获取一种单一的颜色值。
ROLLBACK	ROLLBACK 命令可以中止当前事务处理过程中进行的任何改动。
_SCREEN	_SCREEN 关键字可用作一个限定符, 它能把 FoxPro 主窗口当作一个对象进行处理。
SEC()	SEC() 函数的作用是从一个 DateTime 表达式里面返回对应的秒部分。
SET CLASSLIB	SET CLASSLIB 命令的作用是打开一个 VCX 可视类库文件, 其中包含了相应的类定义信息。
SET COLLATE	在随后的索引和排序操作中, SET COLLATE 命令可用于为字符字段指定一种排列顺序, 从而为国际化应用程序提供支持。
SET CPCOMPILE	SET CPCOMPILE 命令的作用是为编译后的程序指定代码页。
SET CPDIALOG	SETCPDIALOG 命令可以指定在某个程序里面打开一张表格的时候, 是否显示出 Code Page 对话框。
SET DATABASE	SET DATABASE 命令的作用是指定当前的数据库。
SET DATASESSION	SET DATASESSION 命令可以激活指定的表单区。每个区都针对一个表单或者报表定义了一种数据环境。同时可以执行多个区。
SET FDOW	SET FDOW 命令的作用是指定一个星期的第一天是星期几。
SET FWEEK	SET FWEEK 命令的作用是指定用于判断一年内第一个星期的规则。
SET NULL	SET NULL 命令的作用是判断 ALTER TABLE, CREATE TABLE 以及 SQL INSERT 命令是如何支持空值的。
SET OLEOBJECT	SET OLEOBJECT 命令的作用是指定当不能定位一个对象的时候, Visual FoxPro 是否需要到 Windows 注册表进行检索。
SET PALETTE	SET PALETTE 命令的作用是指定是否使用了 BMP 图形的图像调色板。
SET SECONDS	SETSECONDS 命令的作用是指定一个 DateTime 值的时间部分里面是否需要显示秒数。

SET TOPIC ID	SET TOPIC ID 命令的作用是根据主题的关联 ID 指定帮助主题,调用 FoxPro 帮助系统的时候就会打开这个主题。
SETFLDSTATE()	SETFLDSTATE()函数的作用是通过对远程表格创建的本地游标内的一个字段或者记录分配一个字段或者删除状态值。
SQLCANCEL()	SQLCANCEL()函数的作用是请求退出一个正在执行的 SQL 语句。
SQLCOLUMNS()	SQLCOLUMNS()函数的作用是对指定数据源表格内的一系列栏目名以及与每个栏目有关的信息进行处理,把它们恢复成一个 FoxPro 游标。
SQLCOMMIT()	SQLCOMMIT()函数的作用是对一次事务处理进行委托。
SQLCONNECT()	SQLCONNECT()函数的作用是建立与一个数据源的连接。
SQLDISCONNECT()	SQLDISCONNECT()函数的作用是中断与一个数据源的连接。
SQLEXEC()	SQLEXEC()函数可以向数据源发出一条 SQL 语句,语句将在那个数据源内进行处理。
SQLGETPROP()	SQLGETPROP()函数的作用是返回用于一个活动连接、数据源或者附属表格的当前或缺省设置。
SQLMORERESULTS()	假如有多个结果集可用,便可利用 SQLMORERESULTS()函数把另外一个结果集拷贝到一个 FoxPro 游标里面。
SQLROLLBACK()	SQLROLLBACK()函数可以退出当前事务处理过程中进行的任何改动。
SQLSETPROP()	SQLSETPROP()函数可以指定用于一个活动连接、数据源或者附属表格的设置。
SQLSTRINGCONNECT()	SQLSTRINGCONNECT()函数可以通过一个连接串建立与一个数据源的连接。
SQLTABLES()	SQLTABLES()函数的作用是把表格的名字存储到用于一个 FoxPro 游标的数据源内。
SYS(2029)	SYS(2029)函数的作用是返回一个表格类型。
TABLEREVERT()	TABLEREVERT()函数可以放弃对一个缓冲行或者一个缓冲表格或游标进行的改动,并根据磁盘上的当前数据对其进行恢复。
TABLEUPDATE()	TABLEUPDATE()函数的作用是实施对一个缓冲行缓冲表格或游标作出的改动。
THIS	在建立一个对象之前,THIS 关键字提供了对这个对象的一个引用。
THISFORM	THISFORM 关键字在一个表单建立之前提供了对它的一个引用。
THISFORMSET	THISFORMSET 关键字在一个表单集建立之前提供了对它的一个引用。
TTOC()	TTOC()函数可以根据一个 DateTime 表达式返回一个字符值。
TTOD()	TTOD()函数可以根据一个 DateTime 表达式返回一个日期值。
TXNLEVEL()	TXNLEVEL()函数的作用是返回一个数值,这个数值指出了当前的事务处理级别。
UPDATE - SQL	SQL 的 UPDATE 命令的作用是用新值对一张表格内的记录进行更新。
VALIDATE DATABASE	VALIDATE DATABASE 命令的作用是检验表格和索引在当前数据库里面的位置是否正确。
WEEK()	WEEK()函数可以根据一个 Date 或者 DateTime 表达式返回一个编

WITH...END WITH

号,这个编号代表了当时是一年中的第几周。

WITH...END WITH 控制结构可以针对一个指定的对象执行多个命令。

_WIZARD

_WIZARD 系统内存变量内包含了 FoxPro 向导应用程序的名字。

Visual FoxPro 引入了对 FoxPro 语言进行的面向对象的编程语言扩展。包含于这些扩展中有两种类型的类:容器类和控件类。在这儿,容器类就好比一个对话框或者表单,其中包含了其他类型的对象,比如控件对象等等。在 Visual FoxPro 3.0 里,容器对象和控件对象取代了 FoxPro 早期版本里面的控件,另外还加上一些新的对象。下面这张表格列出了一些类名称,它们定义了由 Visual FoxPro 支持的对象的属性、事件以及方法:

类名	说明
CheckBox(核选框)控件	建立核选框控件对象。取代了@...GET 核选框。
Column(栏目)对象	建立网格内的一个栏目,这个网格内包含了正文框。
ComboBox(组合框)控件	建立一个组合框控件对象。取代了@...GET 弹出式。
CommandButton(命令按钮)控件	建立命令按钮对象。取代了@...GET 下推按钮。
CommandGroup(命令组)控件	建立一组命令按钮。
Custom(定制)对象	建立一个用户自定义对象。
EditBox(编辑框)控件	建立一个编辑框。取代了@...GET 编辑框。
Form(表单)对象	建立一个表单容器对象,这是窗口的一种高级表单,它在 Fox-Pro 的早期版本中被定义成对话框使用。
FormSet(表单集)对象	建立一个表单集对象,这是一系列表单的集合,用户利用不同的卡片标签便可选出一张表单。
Grid(网格)控件	建立网格控件,这种控件与 Browse 屏幕是类似的。
Header(页眉)对象	建立一个页眉对象,这是一个网格栏目使用的标题。
Image(图像)控件	建立一个图像控件,这是一种图形化控件,它的作用是显示一幅 BMP 图片。这与@...SAY BITMAP 是类似的。
Label(标签)控件	建立标签对象,这种对象可对事件提供响应。
Line(线段)控件	建立一个线段对象,它能对事件作出响应,并显示出线段。类似于@...TO 命令。
ListBox(列表框)控件	建立列表框控件对象。取代了@...GET 列表框。
OLE 控件	用于建立一个 OLE 控件对象。
OptionButton(选项按钮)控件	建立一个选项按钮对象。取代了@...GET 单选按钮。
OptionGroup(选项组)控件	建立一组选项按钮。取代了@...GET 单选按钮。
Page(页)对象	在一个页帧里面建立一个页。这与表单内的另外一个表单类似。页对象里面可以包含控件对象。
PageFrame(页帧)控件	建立一个容器对象,其中包含一个或多个页对象,这些页对象是一个个重叠起来的。
Shape(形状)控件	建立一个几何形状对象,比如矩形、椭圆以及圆形。类似于@...TO 命令。
Spinner(微调器)控件	建立一个微调器控件对象。取代了@...GET 微调器。
TextBox(正文框)控件	建立一个正文框控件对象。取代了@...GET 正文框。

Timer(计时器)控件
Toolbar(工具栏)对象

建立计时器对象,它能按照指定的周期执行相应的代码。
建立用户自定义工具栏容器对象。

下面这张表格列出了在 Visual FoxPro 3.0 里面得以改进的 FoxPro 命令:

类名	说明
AFIELDS()	这个函数里面又增添了一个可选参数,以便用户指定工作区或者表格的别名。这样一来,用户就能插入针对指定表格的字段信息,正在使用这个表格的工作区可以不是当前工作区。除此以外,数组也进行了扩展,从而提供了新的 Visual FoxPro 扩展字段属性。
APPEND	APPEND 命令新增了一个 IN 子句,利用它可以指定一个工作区。
APPEND FROM	APPEND FROM 命令新增了一个 AS 子句,这样就可以指定附加到 FoxPro 表格内的表格或者文件所用的代码页。
APPEND MEMO	APPEND MEMO 命令新增了一个 AS 子句,这样就可以指定附加到备注字段内的文件所用的代码页。
CLOSE DATABASE	CLOSE DATABASE 命令已进行了修改,它现在能关闭当前打开的数据库容器文件、打开数据库内的表格以及在数据库处于活动状态时打开自由表格。
COPY TO	COPY TO 命令新增了一个新的 AS 子句,利用它可以针对由 COPY TO 创建的新表格或者文件指定使用的代码页。
CREATE TABLE	CREATE TABLE 命令已经进行了修改,它现在能支持附加的 SQL 类子句以及关系的完整性。
DEFINE BAR	下面这四种命令都增加了 FONT 和 SYTLE 子句,利用它们能指定准备使用的字体和字体风格:DEFINE BAR,DEFINE MENU,DEFINE PAD 和 DEFINE POPUP。
DELETE	DELETE 命令现在新增了一个 IN 子句,它可以让用户指定一个工作区。
DELETE TAG	DELETE TAG 命令增加了一个 IN 子句,利用它可以对工作区进行指定。
DOW()	现在支持一个新的可选参数,这个参数可以让用户指定一周的第一天。这是针对软件的国际化支持而提供的。
EXPORT	EXPORT 命令新增了一个 AS 子句,它可以让用户指定由 EXPORT 命令创建的新表格或者文件所用的代码页。
EXTERNAL	EXTERNAL 命令已经进行了修改,用它可以支持附加的文件类型。
GATHER	GATHER 命令增加了一个 NAME 子句,它允许用户用取自一个对象的属性替换字段数据。对象是用 SCATTER 命令来建立的。
HELP	用户现在可以建立自己的 Windows 风格帮助系统,它可以用一个数值关联 ID 来指定帮助主题。DBF 类型的帮助文件也进