



Visual FoxPro 6.0

(中文版)

应用与开发

苏 忠 等编著 / 徐国平 张 治 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

内 容 简 介

Visual FoxPro 6.0(亦称 Visual FoxPro 98, VFP 98)关系数据库管理系统是迄今 FoxPro 系列的最新版本。Visual FoxPro 6.0 继续秉承 FoxPro 系列产品的优点,展示出了更为优秀的、基于 Internet 平台的数据库系统。

本书共有 14 章,2 个附录,内容包括:Visual FoxPro 6.0 概述,数据库的建立,数据操作,索引与排序,查询与视图,表单,报表,菜单设计,OLE,面向对象程序设计基础,使用项目管理器,开发网络应用程序,测试与调试,数据库的 Internet 支持,Visual FoxPro 的新增命令与函数等。较详细讨论了在 Internet/Intranet 中访问 Access/Visual FoxPro 数据库的方法,并重点讨论了 O'Reilly 的 Website 与 Microsoft 的 IIS 两种方法,以及 ActiveX 与 Active Document。

本书内容循序渐进、图文并茂、实用性强,可供数据库管理和应用开发人员、有关院校师生以及培训班作为教材与参考用书。

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,翻印必究。

图书在版编目(CIP)数据

Visual FoxPro 6.0(中文版)应用与开发/苏忠等编
著. —北京:电子工业出版社,1999.9
ISBN 7-5053-4791-8

I. V… II. 苏… III. 关系数据库-数据库管理系统, FoxPro6.0-基本知识 IV. TP311.13

中国版本图书馆 CIP 数据核字(1999)第 03040 号

书 名: Visual FoxPro 6.0(中文版)应用与开发

编 著 者: 苏 忠等

审 校 者: 徐国平 张 治

责任编辑: 施玉新

特约编辑: 苏民生

印 刷 者: 北京朝阳隆华印刷厂

出版发行: 电子工业出版社 URL: <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: 30 字数: 768 千字

版 次: 1999 年 9 月第 1 版 2000 年 4 月第 3 次印刷

书 号: ISBN7-5053-4791-8
TP·2325

印 数: 4000 册 定价: 38.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话: 68279077

前言

FoxPro 关系数据库管理系统在国内流行已久,为众多的用户所熟悉。因此,FoxPro 的每一步发展,总会引起广大数据库用户的关注。1995 年,Microsoft 公司推出了 Visual FoxPro 3.0,可以说是 FoxPro 发展史上的一个里程碑。无论在显示界面还是在内容上,Visual FoxPro 3.0 都比以往版本有着重大的改进。1996 年推出的 Visual FoxPro 5.0,大大增强了其 Internet/Intranet 功能。1998 年 Visual FoxPro 6.0(亦称 Visual FoxPro 98)继续秉承 FoxPro 系列产品的优点,是一个更为优秀的、基于 Internet 平台的数据库系统。其中的许多新颖特点,正为广大数据库用户所关注。帮助读者迅速掌握和使用 Visual FoxPro 6.0,正是撰写本书的目的。

本书力求反映 Visual FoxPro 6.0 的主要特点,对 Visual FoxPro 6.0 的许多细节都有较为详细的描述。本书作者有多年执教 Visual FoxPro 的经验,书中融合了作者的教学体会和应用开发经验,无疑对于全面掌握和编写 Visual FoxPro 6.0 都是有帮助的,无论对于初学者还是熟练用户都将从本书中得到收益。

全书共有 14 章,2 个附录,内容如下:

第 1 章 Visual FoxPro 概述,主要描述自 Visual FoxPro 3.0~6.0 各个版本的主要特点及新增功能; 第 2 章创建 Visual FoxPro 数据库的总体思路及步骤; 第 3 章 BROWSE 窗口及 BROWSE 命令的强大功能及其使用方法; 第 4 章如何对数据表进行排序; 第 5 章创建远程视图和本地视图的方法; 第 6 章表单的概念和创建方法以及在表单中如何使用各种控件; 第 7 章创建报表的步骤及如何使用报表; 第 8 章菜单的概念及如何创建菜单; 第 9 章 Visual FoxPro 中使用 OLE 的方法; 第 10 章 Visual FoxPro 的面向对象程序设计方法(对 OOP 概念、类概念及使用、对象的创建及使用都做了详细的描述); 第 11 章如何使用项目管理器、如何开发应用程序; 第 12 章在网络环境下如何开发 Visual FoxPro 应用程序; 第 13 章程序错误的几种类型以及如何使用调试器; 第 14 章较详细讨论了在 Internet/Intranet 中访问 Access/Visual FoxPro 数据库的方法,这也是本书的重点之一。目前国内这方面的工作正在开始阶段,在本书中则着重讨论了 O'Reilly 的 Website 与 Microsoft 的 IIS 两种方法,以及 ActiveX 与 Active Document; 附录 A 和附录 B 列举了 Visual FoxPro 新增的所有函数和命令。

本书内容循序渐进、图文并茂、实用性强,可供对 FoxPro 2.X 已有初步基础,准备进一步从事 Visual FoxPro 6.0 的数据库管理和应用开发人员,有关院校师生以及数据库应用培训班作为教材与参考用书。

本书的编写工作由徐国平主持,并得到美国 AT&T 公司系统开发工程师徐鹏力的通力协作。参加具体编写工作的有:苏忠(第 1,3,5 章~13 章,第 15~16 章),曹京伟(第 2 章),

郭力平(第4章)等。徐鹏力、曲大成和苏忠撰写了本书第14章。全书由徐国平、张治审校。王小勇、周立峰、赵香等参加了助编和绘图工作。

在本书的编写过程中,曾得到中国 UNIX 用户协会及中国科学院研究生院罗晓沛教授、清华大学侯炳辉教授的指导和帮助,电子工业出版社为本书迅速出版付出了辛勤劳动,一并致谢。

限于作者水平,书中恐有疏漏或不妥之处,敬请读者不吝赐教。

CUUG 应用研究室

第 1 章 Visual FoxPro 6.0 概述

1.1 简介

自 1989 年 Microsoft 公司推出 FoxPro 1.0 以来,由于 FoxPro 系列产品具有简单易学、功能强大、界面友好等特点,一直受到我国广大数据库用户的关注,也是中小型数据库应用系统开发的首选编程语言。1995 年推出的 Visual FoxPro 3.0,更是 FoxPro 系列产品的一次历史性突破,因为它首次在 Xbase 语言里引入了面向对象编程技术,采用了可视化的概念,首次明确提出支持客户机/服务器体系结构,并且彻底更新了“数据库”的概念。Visual FoxPro 3.0 无疑是 FoxPro 系列产品发展的一个重要的里程碑。1995 年在我国计算机界可以称为 Internet 的“元年”,Internet 的急剧发展和普遍应用,对人们的生活和工作都产生了不可估量的影响,从而也促使了许多支持 Internet 的产品的产生与发展。Microsoft 公司在 1996 年 8 月又不失时机地推出 Visual FoxPro 5.0(未命名为 Visual FoxPro 4.0 仅是为了与其他开发语言,如 Visual Basic, Visual C++ 等统一版本号),这一新版本着重增加了对 Internet 的支持,并且首次在 FoxPro 中提出 ActiveX 技术,使得 Visual FoxPro 朝开放式、国际化的方向迈出了更大的步伐。1998 年,Microsoft 公司又隆重推出 Visual FoxPro 6.0,在这一产品中免费提供了 Internet Explorer 4.01, Visual FoxPro 6.0 进一步加强了 Internet 的功能,强化了其面向对象编程技术,并提高了与其他的 Microsoft 产品之间的信息共享能力。

Visual FoxPro 6.0 是 Internet 时代的一个新的数据库产品,它秉承了以往版本的所有优点,使用户使用时备感亲切,菜单界面还是那么友好易用,语法结构还是那么简单易学,学习时不会存在任何心理障碍。同时,Visual FoxPro 6.0 又是一个符合时代潮流的产品,它的界面风格、编程语言的使用等许多方面与 Visual FoxPro 3.0 存在着相似之处,可以把 Visual FoxPro 6.0 认为是 Visual FoxPro 3.0 的增强版。但是,由于 Visual FoxPro 6.0 全面支持 Internet,它有着更大的发展潜力,同时,它包含了许多当今非常先进的技术,通过 Visual FoxPro 6.0 使用这些先进技术开发出来的应用系统在市场上有着很强的竞争力。在我国,由于 FoxPro 系列产品已在广大用户中拥有牢固的历史地位,所以在目前学习、掌握和使用 Visual FoxPro 6.0 正成为广大数据库用户的迫切需要。

1.2 Visual FoxPro 新特点

Visual FoxPro 6.0 有着良好的兼容性,这意味着使用以往版本开发的应用程序在 Visual FoxPro 6.0 中仍然可以正确地运行。但仅仅运行以往版本开发的应用程序并不能真正地体现 Visual FoxPro 6.0 的优越性和强大功能。所以在使用 Visual FoxPro 6.0 这一新版本时,了解它所包含的新功能是很有必要的。

1.2.1 Visual FoxPro 3.0 新增功能

如果你是 FoxPro 2.X 用户,直接使用 Visual FoxPro 6.0 是完全可以的。但是,必须首先了解 Visual FoxPro 3.0, Visual FoxPro 5.0 和 Visual FoxPro 6.0 的新增功能。

- 面向对象编程技术(Object - Oriented Programming)

Visual FoxPro 3.0 的最大特点是使用面向对象编程技术(OOP),用户可以首先构造“类”,通过“继承”可以产生子类,每一个类都包含一系列属性、事件和方法。由类创建的对象几乎可以完成用户的所有要实现的任务。通过“封装”可以把对象内部的复杂性隐藏起来。使用 OOP 方法,通过“继承”使得代码的重用性大大提高,最大限度地减少了代码出错的可能性。

Visual FoxPro 3.0 提倡使用面向对象编程技术的同时,也支持以往版本所使用的结构化程序设计技术。两种技术并不是互相排斥,而是互相补充的。事实上,许多程序代码仍然体现着结构化编程技术。

- 可视化(Visual)编程技术

在使用 Windows 时,“所见即所得”(WYSIWYG)是我们所体会非常深刻的。在使用 Visual FoxPro 3.0 设计一个用户界面时,无须运行程序,就可看到用户的最终界面。这也是 FoxPro 前面冠以“Visual”的缘故。

- 重新定义数据库的概念

在 FoxPro 的以往版本中,数据库就是一个二维表,表中每一行表示一条记录,每一列表示一个字段。数据库仅仅是数据的集合。而在 Visual FoxPro 3.0 里,数据库的概念被彻底更新了。数据库是由若干个表及表与表之间的关系、视图、连接、存储过程等组成的一个集合。以往版本的数据库在这里被称为自由表,自由表是独立于数据库之外而存在的表,而属于某一数据库的表被称为数据库表。数据库表有着与自由表许多不同的属性。例如,数据库表可以定义长表名、可以包含有效性规则等。

数据库概念的重新定义,使得 Visual FoxPro 3.0 为采用面向对象编程技术奠定了重要的理论基础,也使得采用面向对象编程技术成为可能。因为在面向对象编程技术中,“对象”必须有自己的属性和行为,而 Visual FoxPro 3.0 所定义的“数据库”既有自己的属性(如“记录”),又有自己的行为(如“有效性规则”),所以也成为可处理的“对象”。

- 有效性规则(Validation Rule)

Visual FoxPro 3.0 数据库表的有效性规则有两种类型:字段级规则和记录级规则。当为某一个数据库表的某一字段定义了字段级规则时,如果输入或修改该字段值,对应的字段级规则将被激活。它用来检验字段值输入的正确性。记录级规则是与记录相关的有效性规则,当插入或修改记录时被激活,它用来检验记录数据输入的正确性。

有效性规则只适用于数据库表,在自由表里不存在有效性规则。有效性规则是数据库的一部分。

- 触发器(Trieger)

触发器是一个发生插入、更新或删除操作之后所运行记录级事件代码。不同的事件对应不同的动作,例如,插入操作可以激活插入触发器,删除操作可以激活删除触发器等。触发器在有效性规则之后运行,它们常用于检验已建立永久性关系的数据库表之间的数据完整性。

触发器只适用于数据库表,在自由表中不存在触发器。触发器是数据库的一部分。

- 存储过程(Stored Procedure)

存储过程是指存储在数据库里的过程。该过程可以包含任何允许在用户自定义函数中出现的命令和函数。例如,触发器代码就存放在存储过程里。在远程数据源上,存储过程是指在任何 SQL 数据库中被命名的一组 SQL 语句的集合。

存储过程属于数据库的一部分。当打开数据库时,存储过程所包含的过程和函数就立即发生作用。

- 事务(Transaction)

数据库从一个有效性状态到另一个有效性状态的操作集称为事务。在对数据库进行操作时,通常是以一条命令或一个函数作为一个操作单位,这种操作所产生的结果往往是不可恢复的。而事务是以一组命令或函数作为一个操作单位,如果在事务处理过程中发生意外,即可取消事务,让数据库恢复初始状态。

事务处理只适用于数据库操作。

- 本地和远程视图(Locate & Remote View)

视图是一个定制的、虚拟的、可更新的结果集。本地视图是与本地数据源相关联的视图。远程视图是与远程数据源相关联的视图。另外还有参数化视图,它是指根据输入的参数而查询得到的结果集。在实现客户机/服务器应用程序时,视图是一个非常有效的方法。

- 向导(Wizard)

学习和使用 Visual FoxPro 的快捷方法是使用系统所提供的向导。例如,当不知道如何设计一个表时,使用“表向导”可以一步一步地根据 Visual FoxPro 的对话框里所提示的内容进行设计。在向导对话框里,包括了一些需要回答的简单内容,通过若干个步骤即可完成指定任务的操作。

- 重新设计项目管理器

“项目管理器”在 FoxPro 2.X 里已经出现并得到广泛使用。但它的作用仅仅是把一系列文件组织在一起进行编译。在 Visual FoxPro 3.0 里对“项目管理器”进行了重新设计,在“项目管理器”里,可以按不同文件类型设计和组织相关的文件,其中的文件,如:表单、源程序、菜单程序等,还可以单独运行。

- 表单(Form)代替了屏幕(Screen)

每一个应用程序至少都包含一个用户界面。在以往版本里,屏幕(Screen)是设计用户界面的核心部分。但屏幕往往无法响应用户的许多动作。在 Visual FoxPro 3.0 里,以“表单”代替了“屏幕”。“表单”作为一个可处理的对象,有其许多属性、事件和方法。同时,表单又可以包含多个控件,每一个控件通过各自的属性、事件和方法来实现用户指定的动作。“表单”不应被认为仅仅是功能扩大了“屏幕”,它是一个全新的概念,是在界面设计上实现面向对象编程技术的最充分体现。

- 增加了新的数据类型

与 FoxPro 2.6 相比,Visual FoxPro 至少增加了下列几种数据类型:货币型(Currency)、日期时间型(DateTime)、双精度型(Double)和整型(Integer)。所有这些类型的数据都以二进制数据的形式保存在磁盘上(整型是 4 字节的二进制数据,其他数据类型是 8 字节的二进制数据),使用这些数据类型有两大优点:一是保存数据时所占的磁盘空间较小(一个 8 位数

的数据若用数值型保存将占用 8 个字节,而保存为整形只需要 4 个字节)。这样,从磁盘向内存加载数据和索引时,一次可以加载更多的有用数据,从而提高应用程序的性能;二是不需要任何数据转换。Visual FoxPro 内部的整型数据表需将 ASCII 码转换为二进制,而数值型数据在磁盘上保存为 ASCII 码形式,因此,在每次读数据时,都必须将 ASCII 码转换为二进制,存盘时再转换回去。为了处理整型和双精度型以及日期型和货币型数据,Visual FoxPro 增加许多新的命令和函数。

在 Visual FoxPro 3.0 里,增加了一百多条命令与函数,主要是针对新增功能而设计的。比如,对数据库操作的命令与函数、面向对象程序设计的命令与函数等。这些命令与函数大大扩充了 Visual FoxPro 的功能。详细内容请参阅附录 A 或附录 B。

1.2.2 Visual FoxPro 5.0 新增功能

与 Visual FoxPro 3.0 相比,Visual FoxPro 5.0 又增加了以下新功能:

- 增加显示页(Add Visual FoxPro Display Page)

在运行 Visual FoxPro 5.0 时,屏幕首先出现一显示页,从显示页里,可以新建一个项目文件、查找示例程序、查询 Visual FoxPro 联机文档或打开一已有的项目文件。

- 字段的格式和输入掩码(Format and InputMask)

在“表设计器”里,为了统一显示模式,可为字段定义输出格式;为了减少输入错误和提供有效的输入揭示信息,可为字段定义输入掩码。

- 字段映像(Field Mapping)

当在表单里显示表文件的字段时,根据字段类型的不同,其显示格式也不相同,例如,字符型字段在表单里可能是以“编辑框”(editbox)显示其内容。Visual FoxPro 5.0 为每一种字段类型都定义了一种默认的显示类,详见 Visual FoxPro 系统菜单里的“工具(选项)字段映像”。在“表设计器”里也可为所定义的字段定义显示器。详见“表设计器”(参见本书 2.3 节)。

- 外部联接(OUTER JOIN)

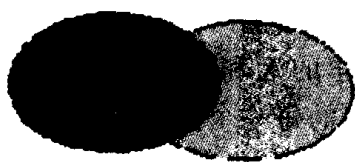
在执行两表 SQL 查询时,通常使用下面的语句:

```
SELECT <field list>;  
FROM <table1> , <table2>;  
WHERE <table1.fieldname> = <table2.fieldname>
```

查询的结果将是满足 WHERE 子句所列条件的所有记录。这种联接称为“自然联接”或“内部联接”(INNERJOINS 或 NATURALJOINS)。然而,在某些情况下,除了需要满足上述条件的记录之外,我们可能还需要这样的记录:尽管父表的记录不是全部满足查询条件,但想把父表的所有记录显示出来,或把子表的所有记录显示出来。但是,上面的语句显然是不能完成如此功能的。Visual FoxPro 5.0 提供外部联接(OUTERJOINS),能满足上述要求。

外部联接一共有三种类型(如图 1.1 所示):

```
LEFT OUTER JOIN  
RIGHT OUTER JOIN  
FULL OUTER JOIN
```



LEFT OUTER JOINs

a. 左联接



RIGHT OUTER JOINs

b. 右联接



FULL OUTER JOINs

c. 完全联接

图 1.1 外部联接的三种类型

左联接(LEFT OUTER JOIN)的查询结果包含了父表的全部记录和子表中满足查询条件的记录。右联接(RIGHT OUTER JOIN)的查询结果包含了父表中满足查询条件的记录和子表的所有记录。而完全联接(FULL OUTER JOIN)的查询结果包含了父表和子表的所有记录。

为了使用新定义的联接, Visual FoxPro 5.0 除了继续支持原来的 SELECT 语句外,还使用了新的语法。例如,执行两表 SQL 查询语句在 Visual FoxPro 5.0 里可改写为:

```
SELECT < field list > ;
FROM table1 ;
JOIN table2 ;
ON table1.field = table2.field
```

综合上述情况,在 Visual FoxPro 5.0 里的 SELECT 语句可书写如下:

```
SELECT < field list >
FROM [DatabaseName!]Table[Local Alias]
[[INNER|LEFT [OUTER]|RIGHT [OUTER]|FULL[OUTER] JOIN]
[, [DatabaseName!]Table[Local Alias]
[[INNER|LEFT [OUTER]|RIGHT [OUTER]|FULL [OUTER] JOIN]
...]
[ON [DatabaseName!]Table[Local Alias].Column Name =
[DatabaseName!]Table[Local Alias].Column Name
```

例如,下面是三表查询的一个程序伪代码:

```
SELECT table1.FieldName,;
table2.FieldName,;
table3.FieldName ;
FROM table3 ;
JOIN table2 ;
JOIN table1 ;
ON table1.FieldName = table2.FieldName ;
ON table3.FieldName = table2.FieldName
```

·支持像素(Pixel)分辨方式

在 Windows 开发中,屏幕分辨率与屏幕位置的测量刻度一般是采用“像素”(Pixel),而 Foxel 方式一般用于交叉平台的应用程序开发。过去,交叉平台意味着 DOS/MAC/Windows/UNIX,现在,交叉平台意味着 Windows 95/Windows NT。

在 Visual FoxPro 5.0 以前的版本里,由于仅支持 Foxel 方式,要检测鼠标指针的像素位置时,必须把鼠标事件,如 MouseUp,MouseDown 或 MouseMove 里的 X 和 Y 坐标作为参数传递给方法代码,但是,鼠标事件并不能提供鼠标精确坐标所需的每一个位置。在 Visual FoxPro 5.0 里,所有的定位函数,如 MROW()或 MCOL()函数,都使用“像素”方式。在这种方式下,就可以精确定位鼠标的位置了。

下面的例子说明在像素方式下精确定位鼠标的位置。

```
LOCAL lnRow, lnCol, lnCount
LOCAL lcFontInfo, lcName
STORE 0 TO lnRow, lnCol, lnCount
lcFontInfo = SPACE(0)

# DEFINE pixel.scale 3
# DEFINE foxel.scale 0

IF ASELOBJ(laForm, 1) = 1
    lcFormName = laForm[1].NAME
    lnRow = MROW(laForm[1].NAME, pixel.scale)
    lnCol = MCOL(laForm[1].NAME, pixel.scale)
    lcFontInfo = GETFONT()
    IF ! EMPTY(lcFontInfo)
        ParseFontInfo(lcFontInfo, @laFont)
        DO WHILE .T.
            lnCount = lnCount + 1
            lcName = "Text" + ALLTRIM(STR(lnCount))
            IF TYPE("laForm[1]."+lcName) = "U"
                EXIT
            ENDIF
        ENDDO
        laForm[1].ADDOBJECT(lcName, "textbox")
        WITH laForm[1].&lcName
            .LEFT = lnCol
            .TOP = lnRow
            .FONTBOLD = "B"&laFont[3]
            .FONTITALIC = "I"&laFont[3]
            .FONTSIZE = VAL(laFont[2])
            .FONTNAME = laFont[1]
        ENDWITH
    ENDIF
ENDIF

PROCEDURE ParseFontInfo
LPARAMETERS tcFontInfo, taFontInfo
```

```
LOCAL lnpos1,lnpos2
```

```
lnpos1 = AT(", ",tcFontInfo,1)
```

```
lnpos2 = AT(", ",tcFontInfo,2)
```

```
taFontInfo[1] = LEFT(tcFontInfo,lnpos1 - 1)
```

```
taFontInfo[2] = SUBSTR(tcFontInfo,lnpos1 + 1,(lnpos2 - 1) - lnpos1)
```

```
taFontInfo[3] = SUBSTR(tcFontInfo,lnpos2 + 1)
```

```
RETURN
```

- 鼠标右键快捷菜单(Right-click shortcut menu)

在 Windows 95 的应用程序设计里,与文本相关的快捷式菜单(shortcut menu)是一个标准。在 Visual FoxPro 3.0 里,尽管可以在 RightClick 事件里为某一个对象编写特定的代码,通过弹出式(pop-up)菜单来实现。但是,这种作法受到一些限制:首先,弹出式菜单不是三维的,与其它应用程序(如 Word, Excel 和 Visual FoxPro)本身的菜单系统在显示方面无差别。其次,弹出式菜单一般是调用 MROW()和 MCOL()函数,由鼠标激活,由于 Visual FoxPro 3.0 仅仅支持 Foxel 分辨方式,要精确定位到指定的位置上是很困难的。Visual FoxPro 5.0 支持像素(Pixel)分辨方式,用 MROW()和 MCOL()函数就可以很方便定位到指定的位置上。创建一个快捷菜单就像创建一个普通菜单一样容易。

要创建一个快捷式菜单,可以在“快捷方式设计器”里设计,详细内容请参阅第 8 章“菜单设计”的相关内容,也可以用编程方式实现,下面就是一个在编程方式下实现快捷式菜单的例子:

- * 菜单程序文件名:devmenu.mpr

```
LOCAL loObject
```

```
loObject = SYS(1270)    && 返回指定位置的对象的参考
```

```
IF TYPE('loObject') = 'O' .AND. ! ISNULL(loObject)
```

```
    IF loObject.Name <> "Screen"
```

```
        RETURN
```

```
    ENDIF
```

```
ELSE
```

```
    RETURN
```

```
ENDIF
```

- * 定义菜单

```
DEFINE POPUP shortcut1 shortcut FROM MROW(),MCOL()
```

```
DEFINE BAR 1 OF shortcut1 PROMPT "调试"
```

```
DEFINE BAR 2 OF shortcut1 PROMPT "类浏览器"
```

```
DEFINE BAR 3 OF shortcut1 PROMPT "其他工具"
```

```
ON SELECTION BAR 1 OF shortcut1 ACTIVATE Window DEBUG
```

```
ON SELECTION BAR 2 OF shortcut1 DO (.BROWSER)
```

```
ACTIVATE POPUP shortcut1
```

在应用程序里加上下面的语句就可以使用快捷式菜单:

ON KEY LABEL RIGHTMOUSE DO devmenu.mpr

- 语法着色(Syntax Coloring)功能

在编写程序代码时,要大量使用 Visual FoxPro 提供的命令、函数、系统变量,以及自定义的一些函数,在 Visual FoxPro 5.0 以前的版本里,系统以统一的显示模式来显示这些代码。用户的代码与系统提供的命令、函数或保留字在屏幕显示方面毫无区别,这种显示模式在程序的阅读与调试过程中无疑会带来一些不便,为了能更好地区分 Visual FoxPro 的命令、函数或保留字与用户自定义的变量和函数,Visual FoxPro 5.0 提供了语法着色(Syntax Coloring)功能。能以不同的颜色显示不同类别的命令、函数和变量。根据显示颜色的不同,就可以区分不同类型的命令、函数和其他内容,用户可以自己调整显示的颜色,具体设置“语法着色”详见 Visual FoxPro 系统菜单里的“工具|选项”。

- 通用命名约定(Universal Naming Convention UNC)

通用命名约定(UNC)在 FoxPro 多用户应用程序开发中是十分必要的。假如,有一台 NT 服务器被称为 NT.ADV.SERVER,该服务器上有一目录为 DATA。在 Visual FoxPro 5.0 以前的版本里,必须为服务器分配一个指定的驱动器号才能访问这个目录。但如果在一个广域网(WAN)的环境里,将不能确定驱动器号是什么,可能只是知道服务器的名字和服务器的有什么类型的文件。在 Visual FoxPro 5.0 里,您将通过使用 UNC,直接访问服务器。例如,有一个名为 TESTDATA 的数据库文件在 NT _ ADV _ SERVER 服务器里的 DATA 目录里,使用下面的语句将可以访问到该数据库文件:

```
OPEN DATABASE \ \ NT.ADV.SERVER \ DATA \ testdata SHARED
```

- 版本信息(Version information)

在 Windows 95“资源管理器”里查看一个文件的属性时,如查看 VFP 目录下的 VFP .EXE 文件,打开“属性”对话框,该对话框有两个选项卡:“常规”选项卡包含了文件的一些固定的信息,如文件创建的日期和时间、文件类型、文件属性(存档、只读、隐藏或系统);“版本”选项卡包含版本号、版权、说明和公司名等。在 Visual FoxPro 5.0 以前的版本里,系统只能为文件自动创建“常规”选项卡所包含的信息。在 Visual FoxPro 5.0 里,可以在建立可执行文件时增加“版本”页所包含的版本信息。当选择建立一个 .EXE 文件时,有一个“版本”按钮,选择该按钮,就可以建立指定的版本信息了(若选择“自动增加”,VFP 将在每一次编译时自动增加版本号)。

使用 FOXTOOLS.FLL 文件里提供的 GETFILEVERSION() 函数,也可以查看 .EXE 文件的版本信息。另外一些文件,如 DLLs 和 FLLs 文件,也提供了版本信息,使用该函数也可查看。

下面的例子通过使用 FOXTOOLS.FLL 文件提供的 GETFILEVERSION() 函数来查看 VFP.EXE 文件的版本信息:

```
DIMENSION laFileVers[12,1]
SET LIBRARY TO HOME() + "FOXTOOLS.FLL" ADDITIVE
GETFILEVERSION(HOME() + "VFP.EXE", @laFileVers)
DISPLAY MEMO
```

LAFILEVERS	Priv	A	examples	
(1, 1)		C	" "	注释文本
(2, 1)		C	"Microsoft Corporation"	公司名
(3, 1)		C	"Microsoft Visual FoxPro for Windows"	内部名字
(4, 1)		C	"5.0"	文件版本号
(5, 1)		C	"VFP"	文件名
(6, 1)		C	"Copyright _ 1996, Microsoft Corporation"	合法版权
(7, 1)		C	" "	合法商业标记
(8, 1)		C	"VFP.EXE"	原始文件名
(9, 1)		C	" "	未用
(10, 1)		C	"Microsoft Visual FoxPro for Windows"	产品名称
(11, 1)		C	"5.0"	产品版本号
(12, 1)		C	" "	未用

• 新的语法(New Syntax)

在 Visual FoxPro 5.0 里,调用函数时不必指定一个存放返回值的变量,甚至不必指定“=”符号来默认存放函数的返回值。例如,下面的函数的书写格式是正确的:

```
inkey(0,"SM")    等待按一个键
```

而这种写法在以前的版本里是错误的。

• 联机文档(Online Documentation)

Visual FoxPro 5.0 提供了完整的联机文档,其内容包括词汇表、概述、用户指南、安装指南、开发指南和语言参考等。用户从联机文档几乎可以得到学习和使用 Visual FoxPro 所需的全部资料。

• 游离视图(Offline View)

视图是与一定的数据源相连接的、可修改的数据子集。但有时会受到一些限制,例如在更新、插入和删除视图时,必须受到所定义的一些规则的限制。在 Visual FoxPro 5.0 里提供了游离视图(Offline View),允许用户在独立于主数据源之外修改或显示数据。在游离视图里工作时与主数据库文件不存在连接关系。但用户可以在修改游离视图的数据后重新建立连接,把修改的内容传给主数据源。

可用 CREATEOFFLINE()函数建立游离视图,游离视图的操作与联机视图是一样的。但不能在“视图设计器”里修改游离视图。必须以编程方式来修改。若要删除游离视图,则可用 DROPOFFLINE()函数。

要把游离视图里已修改的数据传给主数据源时,必须使用 USE 命令里的 ONLINE 子句打开游离视图并与主数据源重新建立连接。如果建立连接成功,则可用 TABLEUPDATE()函数完成对主数据源的更新操作。

例如,下面的例子是用游离视图更新本地表的记录:

```
USE myofflineview ONLINE EXCLUSIVE    打开视图并重新与主数据源连接
BEGIN TRANSACTION
IF TABLEUPDATE (2, .F., "myofflineview")    若更新成功则结束事务
END TRANSACTION
```

ELSE

MESSAGEBOX("更新失败!")

提示出错信息

ROLLBACK

放弃更新

ENDIF

- 提供 Web 浏览器(Microsoft On Web)

Visual FoxPro 5.0 的一个最大的改进就是增加了访问 Internet 的功能。如果您的设备已与 Internet 互连,那么通过系统菜单就可以启动 Web 浏览器,进入 Microsoft Web 节点,从中可以浏览有关 Visual FoxPro 的一些主页。详见 Visual FoxPro 系统菜单里的“帮助 | Microsoft On Web”

- ActiveX 技术

近几年,由于 Internet/Intranet 的迅猛发展,资源共享日益成为一个突出的问题,为了能够更加方便灵活地使用软件资源,Microsoft 提出了 ActiveX 标准。它使得用不同语言开发的应用程序可以在网络中相互操作。ActiveX 是一套技术规范,它本身与开发语言无关。ActiveX 标准主要有三大常用技术:控件(Controls)、文档(Documents)和脚本(Script)。ActiveX 控制在以前也称为 OLE 控件或 OCX 控件。但 OLE 控件或 OCX 控件侧重于单机应用程序的开发和使用,而 ActiveX 更侧重于网络和 Internet/Intranet。ActiveX 文档可以使您能够在 Web 浏览器里看到非 HTML(Hypertext Markup Language 超文本置标语言)格式的文档。如 Microsoft Word,Excel 等文件。ActiveX 脚本支持多种流行的脚本语言,如 Visual Basic 和 Java 脚本。脚本提供了在一个页面中多个对象交互的功能。

Visual FoxPro 5.0 包含了一套 ActiveX 控件,在安装 Visual FoxPro 时一般都会默认安装在 Windows 95 的 SYSTEM 目录或 Windows NT 的 SYSTEM32 目录。可以把 ActiveX 控件加在应用程序里,或通过 OLE 容器控件加到表单里。但有些 ActiveX 控件最早是为其它应用程序创建的,如果在表单里想了解这些 ActiveX 控件的帮助信息时,这些帮助信息要从原始应用程序处显示,而不是 Visual FoxPro。

在 Visual FoxPro 5.0 中,ActiveX 控件必须存在于 OLE 容器控件里。如果要在表单中添加一个 ActiveX 控件,首先要把一个 OLE 容器控件添加到表单里。然后选择“Insert Object”,从中挑选一个 ActiveX 控件,就可把该 ActiveX 控件添加到表单里。

- 调试程序(DEBUGGER)

VFP 5.0 把跟踪窗口(TRACE)与调试窗口(DEBUG)合并为 Visual FoxPro 调试程序。并且扩大了调试范围。除了继续能调试 .PRG, .FXP 和 .MPR 文件之外,还可以调试表单以及方法代码等。调试器的多窗口显示,使得调试信息更加丰富。在调试器里,可以设置事件跟踪序列,也可以根据不同的类型设置断点,跟踪的方法也不再局限于原来的两种(step, over)。

- 新的向导(New Wizards)

在原有向导的基础上,VFP 5.0 扩充两个向导:应用程序向导(Application Wizard)和因特网搜索向导(Internet Search Wizard)。应用程序向导主要用于构造一个 Visual FoxPro 应用程序或项目文件的框架;而因特网搜索向导用于在 Internet 建立一个基于 Visual FoxPro 数据的 Web 页。除了新增两个向导之外,VFP 5.0 把原来的 Upsizing Wizard 分解为 Oracle Upsizing Wizard 和 SQL Server Upsizing Wizard 两个向导。

- 单文档界面(Single Document Interface)和多文档界面(Multiple Document Interface)

Visual FoxPro 5.0 允许建立两种类型的应用程序:SDI 应用程序和 MDI 应用程序。在一个 SDI 应用程序里,用户界面上每一次至多只能打开一个用户文档;而在 MDI 应用程序里,可以同时存在多个用户文档,在 Visual FoxPro 5.0 的“表单设计器”里可以设计三种类型的表单:

子表单(Child Form):在父表单里,不可移动。

浮动表单(Floating Form):属于父表单,可以移动,但不能隐藏在父表单后面。

顶层表单(Top-level Form):独立表单。

各种类型的表单的具体设置请参阅第六章相关内容。

- 更多的示例程序(More Samples and Utilities)

阅读并理解一些示例程序,是学习和掌握编程语言的快捷方法。Visual FoxPro 5.0 提供了范围更为广泛的示例程序。从这些示例程序中可以学到许多开发应用程序的技巧和方法。

- 新增命令和函数(New Commands and Functions)

在 Visual FoxPro 5.0 里,新增的命令有:

CLEAR RESOURCE	清除内存的资源
DEBUGOUT	把表达式的结果输出到调试窗口
DOEVENTS	执行所有等待的 Windows 事件
DROP TABLE	从当前数据库里移去一个表,并从磁盘上删除该表
DROP VIEW	从当前数据库里删除一个 SQL 视图
FOR EACH...ENDFOR	执行一个数组或一个集合每一个元素的命令集
SET DEBUGOUT	把调试结果输出到一个文件里
SET EVENTLIST	把要跟踪的事件指定在调试输出窗口或文件里
SET EVENTTRACKING	启动或关闭事件跟踪,或指定一个文本文件存放事件跟踪信息

新增的函数有:

BINTOC()	把一个二进制数转换为字符串表达式
CREATEBINARY()	把一个字符型数据表达式转换为二进制型数据表达式
CREATEOFFLINE()	创建一个游离视图
CTOBIN()	把一个字符表达式转换为二进制表达式
DODEFAULT()	在一个子类里执行父类的事件或方法
DROPOFFLINE()	放弃游离视图的所有更改,并把视图重新联机
ISFLOCKED()	返回表的锁定状态
ISRLOCKED()	返回记录的锁定状态
LOADPICTURE()	为位图、图标或 Windows 图元文件创建对象参考
PEMSTATUS()	返回属性、事件或方法的属性字
SAVEPICTURE()	从一个图形对象参考里创建一个位图文件
SQLPREPARE()	为 SQLEXEC()函数的远程执行准备一条 SQL 语句

STRCONV()	转换字符串表达式里的各种类型表达式
SYS(1023)	使 HELP 特征模式可用
SYS(1024)	使 HELP 特征模式不可用
SYS(1269)	属性信息
SYS(1270)	返回指定位置上的对象参考
SYS(1271)	返回包含对象实例化的 .SCX 或 .VCX 文件
SYS(1272)	返回对象层次
SYS(1500)	激活系统菜单的一个菜单项
SYS(2029)	返回当前表的类型
SYS(2333)	控制 ActiveX 界面
SYS(3004)	返回本地标识号
SYS(3005)	设置本地标识号
SYS(3006)	设置语言及本地标识号
SYS(3050)	设置前台或后台缓冲内存的大小
SYS(3051)	设置上锁重试时间间隔
SYS(3052)	是否使用 SET REPROCESS 设置去上锁
SYS(3053)	返回 ODBC 环境句柄
TARGET()	返回建立关系的目标表的别名

1.2.3 Visual FoxPro 6.0 新增功能

与 Visual FoxPro 5.0 相比, Visual FoxPro 6.0 则增加了以下新功能:

- Access 和 Assign 方法

在面向对象程序设计中,为某一个对象的属性赋予一个值是经常使用的操作。但是,有些时候并不是所有的赋值都是正确的。例如,对于表单对象(Form)的 Left 属性,若赋予一个负值则是错误的。Visual FoxPro 为了防止出现类似这样的错误,特增加了两个用于检查属性赋值是否正确的方法: Access 和 Assign。

当要查询对象的某一属性值,如把属性值赋给一个变量,或使用问号(?)查询属性值时,系统将会自动调用与之相对应的 Access 方法程序代码。

当要改变对象的某一属性值,如使用“STORE”或“=”命令把一个值赋给某一属性时,系统将自动调用与之相对应的 Assign 方法程序代码。

使用 DEFINE CLASS 命令或使用系统菜单提供的功能可以创建 Access 方法或 Assign 方法。系统提供了两个后缀: _ACCESS 和 _ASSIGN,专门用于创建与之对应的属性的方法程序,请看下面的例子:

```
DEFINE CLASS MyForm AS FORM
    PROCEDURE Left _ Access      && 访问 Left 属性时激活
        MESSAGEBOX("欢迎访问 Left 属性")
    ENDPROC
```

```

PROCEDURE Left_Assign      && 对 Left 属性赋值时激活
LPARAMETERS Left_Value
IF Left_Value < 0
    MESSAGEBOX("Left 属性不能赋予一个负值!")
ELSE
    THIS.Left = Left_Value
ENDIF
ENDPROC
ENDDEFINE

```

Access 方法和 Assign 方法并不是必须同时出现。用户完全可以根据需要使用其中的任一个。但如果对象的属性是只读的(如 ParentClass 属性),则不能使用 Assign 方法。如果不使用系统提供的后缀(_ACCESS 或 _ASSIGN),所编写的 Access 或 Assign 方法将被视为 Visual FoxPro 的标准方法程序。

• Active Document

Active Document 是基于 Windows 的、非 HTML 格式的应用程序。原先只用于 Microsoft Office。目前广泛应用于 Microsoft 的各种产品中。使用 Active Document,用户可以在一个工具上,如 Internet Explorer,从不同的文件源上浏览不同类型的文档。

在 Visual FoxPro 6.0 里,Active Document 是一种新型的应用程序。在这种新型的应用程序里也可运行表单、报表、标签或使用类定义。Visual FoxPro 6.0 专门添加了一个新类 ActiveDoc 用于创建 Active Document 对象,ActiveDoc 对象是 Active Document 应用程序的主程序。

• 组件管理库(Component Gallery)

在 Visual FoxPro 6.0 里,组件管理库是一个容器,它包含了一些软件对象的目录,这些软件对象被称为“组件”。组件管理库可以包含任何的 Visual FoxPro 6.0 的组件,这些组件甚至可以包括本地或远程文档、文件夹、OLE 服务程序(如 Excel, Word 等)、HTML 定位符和文件、.PRG 文件、程序代码段、类、向导、生成器或艺术字等。使用组件管理库可以把库中所包含的组件拖放到指定的对象、项目或应用程序里。在组件管理库里可以改变组件的属性设置。

• 新的向导(New Wizards)

Visual FoxPro 6.0 在原有的向导的基础上,又增加了 5 个新的向导。它们分别是:连接向导、数据库向导、示例向导、Web 发布向导。其中连接向导包括代码生成向导和逆转工程向导。

代码生成向导(Code Generation Wizard)主要用于从一个 Microsoft 可视模型文件(.MDL)里把一个对象模型导入到 Visual FoxPro 里。逆转工程向导(Reverse Engineering Wizard)主要用于把一个 Visual FoxPro 类定义导出到一个 Microsoft 可视模型文件(.MDL)里。

数据库向导(Database Wizard)主要用于创建一个数据库及其所包含的表。它提供了一系列数据库模板供选择,使用不同的模板可以创建不同类型的数据库。使用该向导也可以在数据库表之间建立关联和索引。