



面向对象编程

VISUAL FOXPRO

3.0

彭 棋 曹晓阳 编



成都科技大学出版社

VISUAL FOXPRO 3.0 面向对象编程

彭 祺 曹晓阳 编

成都科技大学出版社

[川] 新登字015 号

责任编辑:杨旭明

封面设计:吴 维 刘 静

内 容 提 要

微软继推出Visual Basic和Visual C++以后,又推出另一力作Visual FoxPro, Visual系列软件在微软原有软件基础上作出了大刀阔斧的改进,其声势已超越了本已深受大家喜欢的Borland公司的编程软件系列。

Visual FoxPro 仍然秉承Visual系列的功能强大,直观好用,面向对象等优点,彻底革新了已大受欢迎,广泛使用的FoxPro数据库软件。

Visual FoxPro 提供了直观设计器(Visual Designers)使得创建Forms, queries, reports, views, classes, tables, databases 和connections 等变得非常容易。而其属性窗口可使你迅速地操作对象。

Visual FoxPro 还提供了向导(Wizards)和建筑师(Builders)。熟悉微软公司产品的朋友应对这两个东西感到亲切。Wizards 使得你能够在软件本身的指引下完成诸如创建forms 和reports, 输入数据和创建查询。而Builders 可帮助你创建复杂的控制。

Visual FoxPro 也提供了与Microsoft office 的更加集成。

另一个重要的方面, Visual FoxPro 支持面向对象编程。通过创建可再次使用的对象,可以迅速创建原型并应用于应用程序。本书提供了Visual FoxPro 所提供的所有类和对象的说明,同时详细说明了所有属性(Property)、事件(Event)和方法(Method)。为节省篇幅和使用上的方便,我们略去了与以前FoxPro 版本相同的命令及说明,仅列出了新增命令和增强了功能的那些命令。

同时需要指出的是, Visual FoxPro 提供了增强的数据库引擎(Enhanced Database Engine),它提供了内置的支持在一个客户——服务器环境中的应用程序的数据共享以及内置的数据字典支持你的生意规则,同时提高了数据操作的速度。

本书针对的是渴望紧跟技术进步步伐的朋友们。随着Windows95的发布,微软公司进一步占领了软件市场,而Windows95确实提供了更为快捷、方便的操作环境,而在其下的Visual系列软件也将更加得以发扬光大,学习这种大家熟悉的数据库产品,并从中学习微软新技术加深对软件趋势的认识,我们认为是可取且有效的。

计算机应用自学教程

VISUAL FOXPRO 3.0 面向对象编程

彭 祺 曹晓阳 编

*

成都科技大学出版社出版

四川石油局印刷厂印刷

新华书店经销

*

开本 787×1092 1/16 印张 14.20 字数 358.4 千字

1995年11月第一版

1995年11月第一次印刷

印数 1—3000册

ISBN7-5616-3055-7/TP·133

定价: 30.00元

目 录

第一章 使用Visual FoxPro	(1)
1.1 共享数据存取编程	(1)
1.1.1 文件中的数据存取访问	(1)
1.1.2 数据的有限访问 (Limiting Access to Data)	(1)
1.1.3 使用Data Sessions (Using Data Sessions)	(2)
1.1.4 Buffering Access to Data	(3)
1.1.5 检测和解决冲突 (Defecting and Resolving Conflicts)	(4)
1.1.6 使用事务 (transactions)	(4)
1.2 工程 (Projects)	(5)
1.2.1 开发多语种应用程序 (Developing International Applications)	(5)
1.3 查询和视图 (Queries and Views)	(8)
1.3.1 查询你的数据 (Querying Your Data)	(8)
1.4 Forms (表单)	(7)
1.4.1 Forms 的建立 (Creating Forms)	(10)
1.4.2 表单的修改 (Modifying a form)	(10)
1.4.3 Forms 的定制	(11)
1.4.4 保存Form	(12)
1.4.5 运行Form	(12)
1.5 菜单和工具栏 (Menus and Toolbars)	(13)
1.5.1 设计菜单	(13)
1.5.2 创建自定义工具栏	(14)
1.6 编程方法	(15)
1.6.1 基础	(15)
1.6.2 在Visual FoxPro 中工作	(16)
1.6.3 面向对象程序设计	(17)
第二章 Visual FoxPro 的设计器	(21)
2.1 类设计器 (Class Designer)	(21)
2.2 联接设计器 (Connection Designer)	(22)
2.3 数据库设计器 (Database Designer)	(21)
2.4 数据环境设计器 (Data Environment Designer)	(24)
2.5 Form 设计器 (Form Designer)	(26)
2.6 菜单设计器 (Menu Designer)	(27)

2.7 查询和视图设计器 (Query and View Designer)	(28)
2.8 报表设计器 (Report Designer)	(30)
2.9 表设计器 (Table Designer)	(31)
第三章 技术参考	(34)
3.1 Visual FoxPro 命名规则	(34)
3.2 数据和字段类型	(34)
3.3 数据容器作用域	(34)
3.4 运算符 (Operators)	(35)
3.5 核心文件集 (Core Event set)	(36)
3.6 Visual FoxPro 基类	(36)
3.7 最小事件集	(37)
3.8 基类最小属性集	(37)
3.9 容器类 (Container Classes)	(37)
第四章 Visual FoxPro 中的类与对象	(38)
4.1 语言参考	(38)
4.1.1 已被废弃的命令和函数	(38)
4.1.2 语音概述	(38)
4.1.3 请按F1 键调用帮助	(39)
4.1.4 使用代码实例	(39)
4.1.5 文档约定 (Document Connections)	(39)
4.2 类和对象参考	(40)
4.3 类的属性	(70)
4.4 事件	(140)
4.5 方法 (Method)	(156)
第五章 新增命令和增强的函数功能	(168)
5.1 Visual FoxPro 新增命令和函数	(168)
5.2 增强功能的命令	(210)
第六章 编程实例	(213)
6.1 对象的拖放	(213)
6.2 显示一系统时钟	(216)
6.3 多页的Form	(219)

第一章 使用Visual FoxPro

1.1 共享数据存取编程

1.1.1 文件中的数据存取访问

共享存取访问 (Shared Access): 当你为了共享使用而打开一个表时, 多个工作站可同时使用同一张表。当你在界面 (interface) 中打开一个表, 可以使缺省的EXCLUSIVE ON 设置失效。使用Visual FoxPro 可明确申明以共享方式打开表。(在一个共享环境中, 可有两种方式访问数据: 从独占文件或共享文件中。)

例如: 在应用当中, 可在命令窗口中键入如下命令得到共享:

```
SET EXCLUSIVE OFF
```

```
USE cMyFile
```

独占访问 (Exclusive Access): 大多数以限定方式打开一个文件的都是独占的。当你通过界面打开一个表时, 它将缺省地以独占方式使用。使用Visual FoxPro 也可明确申明以独占方式打开表。

同样有例:

```
SET EXCLUSIVE ON
```

```
USE cMyFile
```

或更简单地键入USE cMy Table EXCLUSIVE 即可。

在共享环境中, 你可从独占文件中或共享文件中这两种方式下去访问数据。假如你为共享访问而打开表时, 其他用户也可访问该文件。而当你以独占方式打开一个表后, 其他用户就不能存取该文件。由于独占使用使网络中共享数据功能失效, 所以应极少使用它。

1.1.2 数据的有限访问 (Limiting Access to Data)

如果你对文件进行共享访问, 你也必须用锁定表和记录的方式管理数据访问。锁定能提供数据长—短项控制 (both long—and short—term control of data)。Visual FoxPro 提供了自动和手工两种锁定方式。分述如下:

一、自动锁定记录和表的命令

命 令	锁定范围
ALTER TABLE	整个表 (Entire table)
APPEND	Entire table
APPEND BLANK	Table header
APPEND FROM	Entire table

APPEND FROM ARRAY	Table header
APPEND MEMO	Current record
BLANK	Current record
BROWSE, CHANGE and EDIT	Current record and all records from aliased fields in related table Once editing of a field begins
CURSORSETPROP	Depends on Parameters
DELETE	Current record
DELETE NEXT 1	Current record
DDELETE RECORD n	Record n
DELETE of More than one record	Entire table
DELETE-SQL	Current record
GATHER	Current record
INSERT	Entire table
INSERT-SOL	Table header
MODIFY MEMO	Current record when editing begins
READ	Current record and all records from aliased field
RECALL	Current record
RECALL NEXT 1	Current record
RECALL RECORD n	Record n
RECALL of more than one record	Entire table
REPLACE	Current record and all records from aliased fields
REPLACE NEXT 1	Current record and all records from aliased fields
REPLACE RECORD n	RECORD n and all records from aliased fields
REPLACE of more than one record	Entire table and all files from aliased fields
SHOW GETS	Current record and all records from aliased fields
TABLEUPDATE ()	Entire table
UPDATE	Entire table
UPDATE-SOL	Entire table

二、手工加锁

你可以用锁定函数来手工地锁定一个记录或表。可使用如下几个命令: RLOCK(), LOCK(), FLOCK()。一旦你锁定一个记录或表, 切记当可提供访问给其他用户时及时解锁。

1.1.3 使用DataSessions (Using Data Sessions)

DataSessions 属性决定了你能否在分离数据对话中 (Separate data sessions) 即时打开一个表单 (Form)。微软公司将Form 翻译为表单, 为方便起见, 以后Form 不作翻译。

Datassion 属性: 确定一个表单, 表单集或工具栏 (Form, FormSet, ToolBar) 是否能运行在他自己的数据对话和拥有一个离散数据环境 (Separate data enuironment)。具有如下语法:

Object DataSession [=nSession]

设置: nSession

DataSession 属性设置为:

Setting	Description
1	(缺省) 缺省数据对话。
2	私有数据对话。为每一个被创建的Form, FormSet 或ToolBar 实例创建新的数据对话。

注释 (Remarks): 在设计阶段有诼 (design time) 在运行阶段只读 (read-only at run time).

用DataSession 实现多路数据对话
(To enable multiple data sessions)

选择如下选项之一:

在Form Designer 中, 设置Data Session 属性为2。或在命令窗口中, 键入: cFormname. DataSession=2。Data Session ID 数其自身是一个运行阶段属性, 称为DataSession ID.

DataSession ID 属性: 返回数据对话标识符 (ID)

标识为私有数据对话Formset、Form 或ToolBar 的。如果Form、FormSet 或ToolBar 的DataSession 属性设置为1 (缺省的Data Session), 返回缺省的数据对话标识符。其语法为: Object. Datasession ID.

注释: 在设计阶段为只读在运行阶段为可读写。仅仅用于FormSet、Form 或ToolBar 的Datasession 属性设置为2 (私有数据对话)。

你可以使用SET DATASESSIONID 给Data Session ID 属性来改变数据对话。

当你设置Data Session ID 时, 该设置将影响为对象和该对象所包含的所有对象的数据对话的工作。Data-Session ID 属性设置不影响非包含对象或使用CREATEOBJECT () 建立的对象。

改变Data Session ID 属性设置将增加改变到的数据对象参考计数和减少从其改变来的数据对象的参考计数。但是, 如果一个数据对话是由设置Datasession 属性为2 (私有数据对话) 来建立的, 改变Data Session ID 属性设置并不释放初始数据对话, 在这种情况下, 对象必须被释放到自由的初始对话中。

警告: 改变一个对象的DataSession ID 属性设置, 如果该对象包含一个数据边界控制 (data-bound controls) 将导致该控制丢失原始数据源。一般来说, 在对象上使用Data Session ID 不要包含数据边界控制。

1.1.4 Buffering Access to Data

Visual FoxPro 记录和表缓存帮助你保护在单个记录和在多用户环境中的多记录数据上的数据更新和数据维护操作。缓冲区能自动地测试、锁定和释放记录和表。

一旦你使用了缓存, 在你关闭表或使缓存失效前它将一直起作用。

Visual FoxPro 记录和表缓存

一、要访问、修改和写入一个单独的记录时, 请选用record buffering

二、要缓存几个记录的更新时, 请选用table buffering.

使用Record Buffering 要用CURSORSETPROP () 函数。其中, 如要用悲观的记录锁定 (pessimistic record locking), 则要置值:

=CURSORSETPROP ("Buffering", 2)

如要用乐观的记录锁定 (Optimistic record locking) 则置值:

=CURSORSETPROP ("Buffering", 3)

何谓悲观缓存 (Pessimistic buffering): 它指在多用户环境中当你正在修改一个特定的记录或表时防止其他用户进行访问。它为修改特定记录提供了最安全的环境, 但将降低操作的速度, 与其相反为乐观缓存 (Optimistic buffering): 它为一个有效的更新数据的方法, 因为它仅在数据写入时加锁, 因此在多用户环境中使任何单用户独占时间变为最小。当你在远程表 (remote tables) 上使用记录或表缓存时, Visual FoxPro 强制Optimistic locking。

使用Pessimistic 或Optimistic 决定于是一个或几个记录被加锁, 和他们是怎样和什么时候被释放的。

1.1.5 检测和解决冲突 (Defecting and Resolving Conflicts)

FoxPro 提供了检测冲突的函数和其显示的值。其中函数有:

- GETFLDSTATE ()
- GETNEXTMODIFIED ()
- OLDVAL ()
- CURVAL ()

这些函数命令可使你①检测一个改变了的字段(Field) ②检测一个改变了的记录。③决定初始字段值 ④决定当前字段值。

其函数说明将在命令及函数参考手册中详细说明。

1.1.6 使用事务 (transactions)

Transactions 的应用规则:

1. 一个Transaction 由BEGIN TRANSACTION 命令开始由END TRANSACTION 命令结束。如果在一个END TRANSACTION 之前没有BEGIN TRANSACTION 将会产生一个错误。
2. 在一个ROLLBACK 语句之前没有BEGIN TRANSACTION 将产生一个错误。
3. 一个transaction, 一旦开始, 在遇到END TRANSACTION (或rollback) 之前将一直起作用, 甚至在跨程序和函数之间, 除非应用程序终止, 它将导致一个rollback。
4. Visual FoxPro 为包含在transactions 中的数据查询, 先使用transaction 缓冲区的数据缓存, 然后使用磁盘数据。这使得当前最大部分数据被使用。
5. 如果应用程序在transaction 当中终止, 所有操作都将回绕 (roll back), 意即所有在transaction 中的改变都将被取消。
6. 一个transaction 工作仅在一个数据库容器 (container) 中。
7. 如果覆盖一个已存在的索引文件或任何一个产品索引 (production index) 文件被打开, 你就不能使用INDEX 命令。

1.2 工程 (Projects)

1.2.1 开发多语种应用程序 (Developing International Applications)

一、使用代码页面 (code pages)

1. 检查DBF 文件中的代码页面标记。

你可以在打开文件后用CPDBF () 函数判定一个DBF 文件是否具有代码页面标记,或在打开文件时用Visual FoxPro check.

采用如下步骤可自动地检测代码页面:

(1) 从工具菜单中 (Tool menu) 选择Options.

(2) 单击Datasheet tab.

(3) 为代码页面检测框 (code page check box) 设定提示, 假如还没有设定的话。一旦你设定了提示, 将是永久性的, 除非你设定Use options 直到退出check box.

提示: 除了为Code page check box 设定提示以外, 你可以使用SET CPDIALOG 命令来检测代码页面。

2. 增加代码页面标记。

如果你使用的是以前版本FoxPro 的DBF 文件, 这个文件就没有code page 标记, 没有这个标记, 该文件就不能被正确显示。当你打开文件时, 你就能看到它是否具有code page 标记。假若没有, 你可以增加一个。

采用如下步骤将可以增加一个code page 标记。

(1) 确信自动页面代码检测 (automatic code page checking) 已被激活。

(2) 打开文件。

在The Code page dialog box 中。

(3) 选择适当的code page.

(4) 浏览该文件看你是否已赋予了正确的code page.

如果有些数据看不到, 或不能识别某些数据, 则code page 是不正确的。

(5) 如果code page 不正确, 在Visual FoxPro MISC 目录中使用CPZERO 程序清除code page 标识。

(6) 重复以上过程直到code page 正确。

3. 清除代码页标记。

如果一个DBF 文件不能被正确显示, 它可能有一个错误的code page 标记。你可以用Visual FoxPro MISC 目录下的CPZERO 程序清除这个Code page 标记。运行CPZERO 设置Code page 为0, 意味着没有。其使用语法为:

DO CPZERO WITH Filename

附注 (Note): 当你清除一个数据库文件的Code page 标识时, 文件中的数据不会改变。如果要改变数据的code page, 你必须为该文件标以适当的code page 即用正确的code page 标识文件, 也就是2. 中所述的方法。

4. 改变一个DBF 文件的代码页。

一旦一个DBF 文件具有了Code page, 除非你拷贝该文件否则你不能改变其Code page。

改变一个DBF 文件的Code page。

改变一个DBF 文件的Code page 的方法:

(1) 使用COPY TO 命令。

例如: 拷贝TEST.DBF 到TEST866.DBF, 同时将Code Page 改变为866, 可使用如下命令:

```
USE TEST.DBF
```

```
COPY TO TEST866.DBF AS 866
```

当COPY TO 完成后, 在结果文件中的数据将具有新的code page。

提示: 使用COPY TO 而未带AS 短语则生成的表具有当前Code page。

5. 指定TEXTFile 的Code page。

如果你遗忘了文本文件的code page, 而你的文本文件又不是你工程的一部分, 就不能判定code page. 因为文本文件不像DBF 文件, 它没有代码页标记。记住文本文件的code page 最好的办法是将文件加入到工程中。其步骤是:

(1) 打开Project Manager。

(2) 选择你想要指定代码页的文本文件。

(3) 在Project menu 中, 选择Project Info。

(4) 在Project Window 中, 单击Files tab。

(5) 使用鼠标右键, 单击所选的文件。

(6) 在子菜单中, 选择代码页。Visual FoxPro 将显示代码页对话框。

(7) 选择合适的代码页。

如果你知道一个文本文件的代码页, 你就可以用适当的Visual FoxPro 命令中AS 短语来指定它。在为文件输入和附加中 (import and append), 你可以用IMPORT 或APPEND 命令来指定代码页。在查询、命令和其他你的计算机中已有的文本文件中改变代码页, 可以使用MODIFY QUERY, MODIFY COMMAND 和MODIFY FILE 命令。

6. 判定工程文件的代码页。

将文件加入到工程中后, 你可以用显示与该文件相关的工程信息来判定其代码页。其步骤如下:

(1) 打开Project Manager。

(2) 选择你想知道代码页的文件。

(3) 在Project menu 中, 选择Project Info。

当你从工程中建立了一个应用, Project Manager 将自动将工程中的文件集成, 而不管其中包含了多少不同的代码页。结果应用中将有当前代码页。

附注(Note): 当你将DBF 文件加入到工程中, 你不必为文件指定代码页, 因为Visual FoxPro 会自动从文件代码页标记中判定其代码页。然而, 当你将一个文本文件加入到工程中时, 你必须指定该文件的代码页, 因为Visual FoxPro 不能自动判定其代码页。

二、Working with International Data

1. 输入国别设定字符。

如果你的键盘没有国别设定字符键, 你可以使用由Windows 提供的字符映射表(Character map) 来输入这些字符, 或是通过按住ALT 键同时键入数字来输入。

其步骤为:

(1) 从字符映射表中拷贝字符, 将其粘贴到你文档中。

(2) 按住ALT 键然后键入适当的3 位ASCII 码。提示(TIP): 字符映射表的状态棒(status bar) 显示了映射表上每一个字符相连的组合键图。

2. 数据的翻译保护 (Preventing Translation of Data)

在有些情况下, 你不想自动翻译代码页。一般来说, 如果一个字符字段包含了一个已加密的口令, 你就不愿意Visual FoxPro 自动地翻译该口令, 因为这样做就会改变它。在字符和MEMO 字段中防止数据翻译其步骤如下:

(1) 打开包含表的工程。

(2) 选择表。

(3) 选择修改按钮 (Modify button)。则Table Designer 将会出现。

(4) 选择你要保护的数据的字段。

(5) 从Type 列表中, 选择 “Character (binary)” 即字符字段, 或者 “Memo (binary)” 即Memo 字段。

(6) 选择OK, 然后选择Yes 按钮确认其修改。

(7) 使用DISPLAY STRUCTURE 命令显示表的结构以检验其修改。

三、使用排序次序 (Sort order)

- 1. 了解排序次序
- 排序次序 (略)
- 2. Sort Order Guidelines
- 3. Specifying Sort Order

四、界面参考 (Interface Considerations)

1. 创建消息 (Creating Messages)

当你在应用程序中创建消息, 英文文本字符串通常比与此相当的其它语言文本字符串要短。下面的表显示了基于他们初始长度, 字符串平均的增长率 (增长级数)

English length (in characters)	Additional growth for localized strings
1 to 4	100%
5 to 10	80%
11 to 20	60%
21 to 30	40%
31 to 50	20%
over 50	10%

2. 使用图标 (Icon) 和位图 (Bitmap)

如果使用正确, 图标和位图应该是用户界面中重要的部分。然而, 图标和位图的含义可能比词语 (Word) 的含义来得含糊。因此, 使用图标和位图要遵循以下规则。

(1) 使用全球公认的图象 (images)。例如: 使用信封 (envelope) 代表邮件 (mail), 而不要用邮箱 (mailbox), 因为它不是一个通用的符号。

(2) 使用人文化敏感的图象 (Culturally sensitive images)
例如, 避免使用宗教性的符号和动物图象。

(3) 位图中避免使用文字，因为增加文字将产生一个问题，正好它可处于界面的其他部分。

(4) 避免使用隐含、俚语、幽默、过份的语言和种族陈规。

(5) 如果你要画男人和女人，要确信他们的性别角色是合适的，同时其人体的姿态和图象在目的文化中是适当的（符合当地文化习惯的）。

(6) 正确使用色彩。例如：避免使用易于与国旗和政治运动相混淆的色彩。

如果你不能确认一个图标和位图是正确的话，最好向当地的人请教一下如何设计其应用。

1.3 查询和视图 (Queries and Views)

1.3.1 查询你的数据 (Querying Your Data)

一、建立一个查询。

二、为查询选定字段 (Selecting Fields for the Query)

在你运行一个查询以前，必须选择一些字段包含在结果中。在某种情况下，你可能想要在表或视中使用所有的字段。在其他时间，你又可能想要将查询集中在几个有限的选择字段中，例如那些你想要包含在报表 (report) 中的字段。

排序和组合 (sort and group) 你的查询的结果一定要包含你想要使用的任何字段，当你选择了字段，就能设置顺序以它出现在你的输出中。

为输出字段设置顺序：在所选的输出箱 (Out put box) 中出现字段的顺序决定了你查询输出的信息列的顺序。如果要改变该查询输出的列序 (Column order) 则拖动移动框 (mover box)，定位到字段名的左端，上或下端。在 Query Designer 的下端窗格 (lower pane) 中使用字段标签去选择你想要包含在查询结果中的字段。

使用 Query 和 View Designer 去建立和修改查询和视图。当 Query 和 View Designer 是活动的，FoxPro 将显示查询菜单和查询工具栏和视图工具栏 (Query and view tool bar)。

在 designer 的上端窗口中显示着查询或视图的表。每一个表被一个大小可变的窗口表示，其中列着表的字段。designer 利用显示表间连接字段的线条表明连接条件。你可以通过拖动表间的字段来建立连接条件。双击一条线可显示该连接对话框 (dialog box)，在此可编辑条件。

在 Query 和 View Designer 下端窗格中包含多个标签—Selection Criteria, Fields, Order By, 和 Group By。View Designer 有一个 additional tab, Update Criteria。

如果要为查询输出增加一个字段可用：

选定字段名并选择 Add，或者拖动字段名到所选的输出框 (output box)。如果要为一个查询立即增加所有要用的字段，可用：

选择 Add All，或者拖动表上端的通配符 (星号, asterisk) 到所选的输出框中。

三、选定你需要的记录 (Selecting the Records You Want)

找到你需要选定的记录是决定查询结果的关键步骤。通过 Query Designer 中的选定条件标签 (Selection Criteria tab) 你可以生成一个选定语句公式来告诉 Visual FoxPro 要查找和检索 (retrieve) 的记录。

指定条件选定 (Selection Criteria) 的步骤如下：

(1) 从字段名列表中，选择一个你需要使用的字段作为选择记录的基础。附注：你不能

使用General 或Memo 字段作为条件选择。

(2) 从条件表中, 选择要比较使用的类型。

(3) 在例子文本框 (Example text box) 中键入比较条件。

- 不要在引号 (quotation marks) 边紧跟字符串, 除非查询中的表的字段名与该字符串相同。

- 不要在大括弧 (braces) 中紧跟数据。

- 在逻辑字母 (logical literal) 前后加上句点如 (. T.)

- 如果你在查询的表中输入字段名, Visual FoxPro 将其视为一个字段。

(4) 如果你想要在字符数据搜索中忽略大小写, 请选定Case 按钮。

如果你要将逻辑操作符求反, 请选定Not 按钮。例如, 如果想要在除了华盛顿外的所有地区寻找顾客, 请用下面例子中的选择表达式:

Customer. region Not like WA.

你也可以增加多重条件进一步调整搜索 (tune your search) 即增加OR 条件 (或条件)

四、创建查询结果排序 (Ordering Query Results)

如果将被排序的记录或列 (records or rows) 出现在你的查询的输出中时, Ordering 决定了其顺序。例如, 你能够按州 (state) 后再按城市 (City) 来排序, 或在基于未完成的帐务平衡 (outstanding account balance) 基础上按升序排序。

使用Order By 标签可以为你的查询设定排序顺序。该排序顺序 (sort order) 决定了将出现在你的查询输出中的列和记录的顺序。

首先从Selected Output box 中选定你想要使用的字段同时将他们移动到Ordering Criteria box 中。然后按你想要其出现在查询结果中的顺序来组合字段。

设定条件顺序 (ordering criteria):

(1) 在Selected Output box 中, 选定一个字段名。

(2) 选择Add。

清除条件顺序 (Ordering criteria):

(1) 选定一个或多个你想要清除的字段。

(2) 选择Remove。

五、创建查询结果组 (Grouping Query Results)

Grouping 合并或采集相似的记录到一个记录中, 因此你能够在基于记录组的基础上完成计算。例如: 你可能需要找出所有订单 (orders) 的和到一个指定的域 (region), 除了一个个地在记录中查找外, 你可以从同一个域中将所有记录作为组合而为一个记录, 并从该域中得到所有订单的和。要控制怎样创建记录组, 请使用Query Designer 中的Group By 标签 (tab)。

Grouping 在与一个求和函数共同使用时极其有用, 例如与SUM, COUNT, AVG 等等。

设定grouping options:

(1) 在Fields tab 中, 选定Function/Expressing list。

(2) 选定一个求和函数 (aggregate function) 和一个字段名。

(3) 选择Add 将表达式加入到Selected Output list 中。

(4) 在Group By 标签中, 加入你想要组合结果的字段。

你也可用Set Selection criteria 到组合结果中 (grouped results)

六、Directing the Results of Your Query (指导你的查询结果)

你可以将你的查询结果送到不同的地方 (destination)。如果你没有选定目的地, 你的查询结果将被显示在Browse window 中。

选择一个查询目的地:

(1) 从Query toolbar 中, 选择Query Destination。

(2) 在Query Destination dialog box 中, 选择一个输出目的地, 如果需要的话, 填上任何附加的字段。

使用下面的表来选择查询输出的目的地。

If you want to	Choose this output option
在一个Browse Window 中显示查询结果	Browse
在一个由你命名的临时只读表中保存你的查询结果	Cursor
将你的查询结果作为一个由你命名的表保存起来	Table
生成一个MS Graph 可用的查询结果, 一个包含Visual FoxPro 的分离应用程序。	Graph
在Visual FoxPro 主窗口中或在当前可活动窗口中显示查询结果。	Screen
将输出送入report 文件中 (.FRX)	Report
将输出送到label 文件中 (.LBX)	Label

七、运行一个查询 (Running a Query)

当你完成了查询设计和指定了查询要输出的目的地, 你可以通过选择Run 按钮开始这个查询。Visual FoxPro 执行由你用Query Designer 建立的SQL select statement, 并将其结果输出到你指定的目的地。如果你不选一个指定目的地, 其结果将显示在Browse Window 中。

运行一个查询: 在project Manager 中选定一个查询名, 然后选择Run。

1.4 Forms (表单)

在Forms 中显示你的数据 (Displaying Your Data in Forms)

1.4.1 Forms 的建立 (Creating Forms)

在你的表和机中显示字段和记录, 并且通常包含定位控制来帮助你在记录和记录间移动, 你可在Visual FoxPro 用以下任何方式建立forms。

一、使用Form Wizards 来建立即建即用 (ready-to-use) 的forms

二、在Form 菜单上选择Quick Form 建立一个简单的form, 你可以通过增加自己的控制来定制一个form

三、使用Form Designer 来修改已存在的form 或另造一个form

Form Wizards 的使用方法: 用Form Wizard 生成一个Form, 无论何时你想建立一个新的form, 都可使用一个form wizard 来帮助建立。该向导 (Wizard) 会提一系列的问题, 然后基于所答来建立一个表, 你也可从几个不同风格的选项中进行选择, 同时在建立此form 前可预览。

具体步骤如下:

(1) 在Project Manager 选择文档标签 (Documents Tab) 和表单 (Forms)。

- (2) 选择New。
- (3) 选择Wizard 按钮。
- (4) 选定你想建立的Form 的类型。
- (5) 跟随Wizard 窗口中的指令进行。

你用Form Wizard 建立的Form 将具有一个标准的定位按钮集，所以你就可在该Form 中显示不同的记录，寻找记录等。

1.4.2 表单的修改 (Modifying a form)

如果Wizard 或builder generated forms 不太适合需要，你能用Form Designer 来修改，使用Form Designer 易于移动和调整大小控制，拷贝和删除控制，对齐方式 (Align) 控制以及修改标签顺序 (Tab Order)。

修改一个已生成的Form 的步骤：

- (1) 在Project manager 中选定文档标签。
- (2) 选择Forms 图标并选定Form 的名字。
- (3) 选择Modify。

1.4.3 Forms 的定制

使用Form Designer，你可用以下方式来定制Form。

一、改变文本字体和字体大小

在Properties Window 中使用字体属性，你可修改文本文件的风格和大小，该文本出现在你已加入Form 的控制当中。你可从所用系统支持的可选字体风格和大小中进行选择，在Form 中改变文本、字体属性如出现在Properties window 中的Font Name、FontSize 和FontBold

在Form 中改变文本的方式如下：

- (1) 在Form Designer 中选择你想改变的控制。
- (2) 在Properties Window 中下拉对向列表 (Object list) 选定你要修改的对象。
- (3) 定位到需改变的字体属性，设置其值。

二、在Forms 中增加形状和线条

可在Forms 中增加形状和线条来定制它们。例如：你可能想用分割线来分离控制组或在box 中装入相关控制来使其分离。

1. 在form 中加入线条

- (1) 在Controls 工具栏中选择Line 按钮。
- (2) 在Form 中拖放来划线。

你可象其它控制那样来改变线条粗细和移动线条，或在Properties Window 中用LineSlant 属性来改变线条倾斜度。

2. 在Form 中增加形状 (Shape)

- (1) 在Control 工具栏中选择shape 按钮。
- (2) 在Form 中拖放来生成一个形状。

你可用Properties Window 中的Curvature 来改变形状的类型。

三、在Form 中加入图形

你可将图形加入Form 的背景中或在Form 的一个方框中加入图片。

在Form 中加入图形的方法:

(1) 选择以下之一:

- 选定Form 来改变Form 的背景
- 从Controls 工具栏中选定一个Image control, 将它放在Form 上, 以此将图片加入Form 中。

(2) 在properties Window 中, 选定Picture property。

(3) 单击三点按钮来显示Open 对话框

(4) 定位位图或其它你想要使用的文件。

(5) 选择OK

四、设置Form 的前景色和背景色

使用Palette 工具栏来改变Form 的前景色或背景色, 以及其中的控制。

1. 设置Form 的前景色、背景色及控制。

(1) 从View 菜单中选择Toolbars, 选定Palette。

(2) 选择前景色按钮或背景色按钮。

(3) 选定控制或你要改变的Form

(4) 在Palette 工具栏中, 选定你想使用的颜色。

(5) 可选择Other color 来察看更多的颜色。

2. 生成一个自定义的色彩。

(1) 选择Other color.

(2) 在Color 对话框中, 选择Define Custom Colors 来显示Custom Color Selector.

(3) 选定一个Custom Color 方框, 再在Custom Color Selector 中单击就可选定所需颜色。

(4) 选择Add to custom Colors 在你的调色板中增加新颜色。

(5) 选择OK。

1.4.4 保存Form

在Form 运行之前先要保存它。假如Form 未存盘之前你要运行Form 或关闭Form Designer, Visual FoxPro 将提醒你保存或取消你所作的修改。

保存Form 的方法:

在Form Designer 中, 从File 菜单中选择Save。

1.4.5 运行Form

你可在界面中或程序代码中直接运行Form。

交互式地运行Form。

以下几种方式来运行你设计的Form, 最简单的方法就是在Form Designer 工具栏中单击Run 按钮。

(1) 在Project Manager 中选定Form, 然后选择Run;

(2) 使用DO FORM 命令。

你也可使用选择Do from the Program 菜单来运行Form, 在Type 方框的List Files 中选择Form 类型, 在其中选定一个所需Form, 选择DO。