
地理信息系统二次开发教程

--组件篇

清华大学出版社

第 1 部分 使用 MapBasic 开发 GIS

第 1 章 MapBasic 开发环境

MapInfo 在地理信息方面的应用，为公众提供了一个良好的途径。它强大的桌面化地理信息处理功能，灵活的 OLE（动态链接与嵌入）和 ODBC（开放式数据库连接）远程访问特性，给每个使用该软件的用户留下了深刻的印象。不仅如此，为了方便用户的应用开发，也为了 MapInfo 的普及推广，MapInfo 还提供了能让用户开发和使 MapInfo 自动化的 MapBasic 语言。该编程语言与 Microsoft 的 Quick Basic 和 Visual Basic 有许多相似之处。

由于 MapBasic 是 MapInfo 产品的一部分，因此熟练掌握和使用 MapInfo Professional 软件和理解 MapInfo 相关技术是成功使用 MapBasic 进行二次开发的关键。因此，本章首先将介绍 MapInfo 的相关技术，然后再介绍 MapBasic 的开发环境。

1.1 MapInfo 相关技术

1.1.1 MapInfo 公司及其产品

MapInfo 公司 1986 年成立于美国 Troy（特洛伊）市。自成立以来，该公司致力于为用户提供先进的数据可视化、信息地图化技术，并将这些技术与主流业务系统集成，提供完整的解决方案。MapInfo 吸取了传统 GIS 的精华，并借助于计算机技术的发展，及时将 GIS 概念从中大型计算机的专用工作站上介绍至普通 PC 上，开创了崭新的桌面地图信息系统。MapInfo 公司的主要系列产品有如下几个：

（1）MapInfo Professional 是 MapInfo 公司主要的软件产品，它支持多种本地或者远程数据库，较好地实现了数据可视化，能生成各种专题地图。此外还能够进行一些空间查询和空间分析运算，如缓冲区等等，并通过动态图层支持 GPS 数据。其界面如图 1.1 所示。

（2）MapBasic 是在 MapInfo 平台上开发用户定制程序的编程语言，它使用与 BASIC 语言一致的函数和语句，便于用户掌握。通过 MapBasic 进行二次开发，能够扩展 MapInfo 功能，并与其他应用系统集成。

（3）MapInfo ProServer 是应用于网络环境下的地图应用服务器，它使得 MapInfo Professional 运行于服务器端，并能够响应用户的操作请求；而客户端可以使用任何标准的 Web 浏览器。在服务器上可以运行多个 MapInfo Professional 实例，以满足用户的服务请求，从而节省了投资。

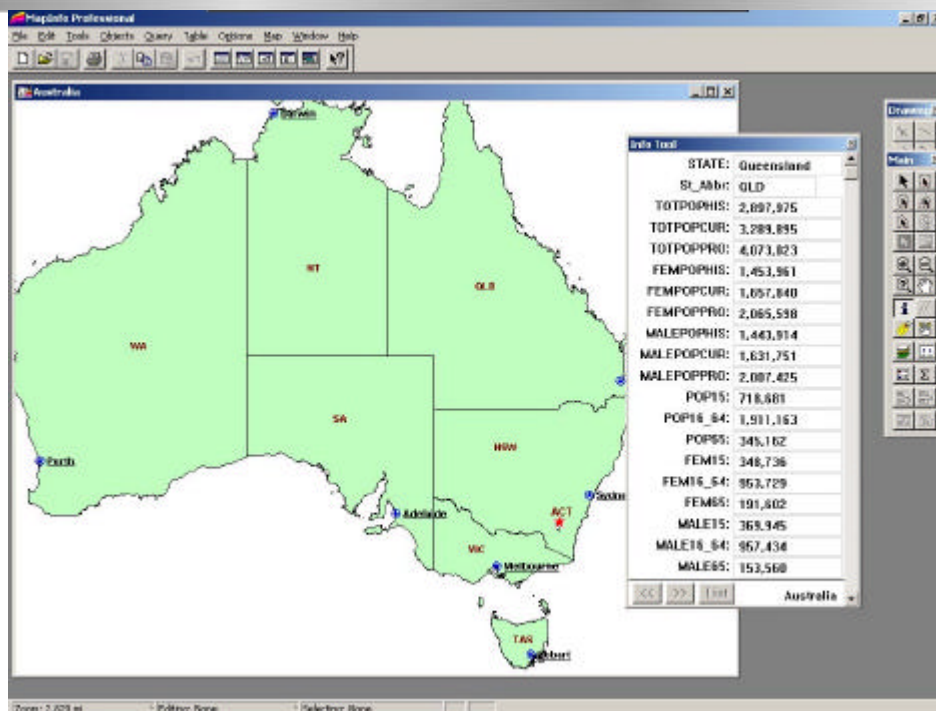


图 1.1 MapInfo Professional 用户界面

(4) MapInfo MapX 是 MapInfo 提供的 OCX 控件。

(5) MapInfo MapXtrem 是基于 Internet/Extranet 的地图应用服务器，它可以帮助配置企业的 Internet。

(6) SpatialWare 是在对象——关系数据库环境下基于 SQL 进行空间查询和分析的空间信息管理系统，在 SpatialWare 中，支持简单的空间对象，因而支持空间查询，并能产生新的几何对象。在实际应用中，一般使用 SpatialWare 作为数据服务器，而 MapInfo Professional 作为客户端，可以提高系统开发效率。

(7) Vertical Mapper 提供了基于网格的数据分析工具。

1.1.2 MapInfo 空间数据的拓扑关系模型

空间数据的拓扑关系模型是地理信息系统的基石。当前商用地理信息系统中，主要采用 POLYVRT 结构和“空间实体+空间索引”两种模型。

MapInfo 采用称作“空间实体+空间索引”的拓扑关系模型。20 世纪 80 年代中后期出现的商用地理信息系统，尤其是桌面地理信息系统大多采用这种方式，包括 ESRI 的 ArcView。而采用 POLYVRT 模型的系统主要是以 Arc/Info 为代表的一些专业 GIS 产品。

“空间实体+空间索引”模型的基础是“空间实体”。空间实体是地理实体的抽象，主要包括点、线、面三种类型。每个空间实体对象都维护着自己的所有属性。多个空间实体组成一个图层。

“空间实体+空间索引”模型的空间查询功能是通过“空间索引”技术来实现的。空间索引的目的是对给定的空间坐标，能够以最快的速度搜索到坐标范围内的空间对象。MapInfo 采用 R-Tree 技术将空间实体的最小外接矩形 (MBR) 存储在索引中，并按从大到小的顺序进行索引搜索。建立了空间索引，就能快速地进行空间分析了。



POLYVRT 的优点是多个不同的对象可以共用相同的结点,从而节省了存储空间,但由此产生的问题是其结构更加复杂,数据的编辑和维护比较困难。而“空间实体+空间索引”模型虽然会造成公共结点的重复存储,但是其结构化的实体模型使得对某个对象的更改不会影响到其他对象的定义,从而大大增强了空间数据的可维护性。

另外,基于 POLYVRT 结构的空間数据组织不能被规范为关系模式,而基于“空间实体+空间索引”模型的空間数据组织可以规范为“实体-关系”模型,从而可以与关系数据库系统结合,在关系数据库内实现对空间数据的查询、分析和处理等操作。

1.1.3 MapInfo 的技术特点

(1) 以表 (Table) 的形式组织信息

每一个表都是一组 MapInfo 文件,这些文件组成了地图文件和数据库文件。为使用 MapInfo,就需要有组成表的用户数据和地图文件。这些文件可以来自 MapInfo,也可以由用户创建。用户要想在 MapInfo 中工作,就必须打开一个或多个表。

MapInfo 通过表的形式将数据与地图有机地结合在一起。当用户在 MapInfo 中打开数据文件时,MapInfo 将创建一个表。这个表至少由两个独立的文件组成,一个是包含数据结构的文件,另一个是包含原始数据的文件。一个典型的 MapInfo 表将主要由 *.tab、*.dat、*.wks、*.dbf、*.xls、*.map、*.id、*.ind 格式的文件组成。

(2) 图形对象

MapInfo 内置的数据库管理系统是一种关系型数据库管理系统,也就是说它也用二维表组织数据。与其他关系型数据库不同的是表结构中除可包含常用类型的属性列外,还引入了一个图形对象列 (OBJ 列),用于存储图形对象 (如线、区域等)。MapInfo 提供了许多图形对象的操作接口,利用这些接口可以生成和处理所需要的各种图形。

(3) 地图图层化

MapInfo 是按图层组织计算机地图的。每个图层包含整个地图的一个方面。例如,第一个图层包含省边界,第二个图层表示省府的符号,第三个图层由标注文本组成,把它们层层叠加就形成了一幅完整的地图。在创建每一个图层时,都要为其建立一张表,MapInfo 就是通过这种方式使表与地图之间建立了联系。

也就是说,MapInfo 是以表的形式来进行管理的,每个表一般包含两部份:地图部分和数据库 (属性) 部分。

(4) 专题地图

提供多种数据可视化的专题地图,能将数据库中的信息进行直观的可视化分析。使用专题渲染在地图上显示数据时,可以清楚地看出在数据记录中难以发现的模式或趋势,为用户提供决策依据。专题地图包括范围值、点密度、柱状图、等级符号、饼图和独立值 6 种形式。

(5) 内置 ODBC

MapInfo 内置 ODBC,支持各种关系型数据库,支持 SQL 查询,从而保证了对原有数据库的沿用和对远程数据库的访问。具备空间查询的扩展功能 (如缓冲区、叠加等),采用数据仓库的最新技术 OLAP (Online Analytical Processing) 的联机事物处理,为应用程序实现图形查询和表查询提供了强大的手段。

(6) 支持多种数据格式及其转换

MapInfo 既支持数字化仪的图形输入方式,直接生成矢量图,也支持目前市场上流行的图形图像格式。可接受 AutoCAD 的 DWG 和 DXF 标准文件格式,还可通过 MIF 及 MID 文



件与其他软件建立数据的接口。其中 MIF 文件内保存有图形信息，MID 文件内保存有图形的属性信息。MIF 及 MID 文件通过 MapInfo 的菜单命令 Import 和 Export 来输入和产生。

(7) 二次开发工具 MapBasic

作为一个系统软件，MapInfo 提供了可以将其所有的功能用程序来驱动的方法，内置标准的二次开发工具——MapBasic。MapBasic 不仅与大众化的 Basic 语法一致，具有基本的常用函数集（计算、字符串处理、文件 I/O、DLL 调用等），而且可以利用 MapBasic 语言所提供的函数、过程和语句命令来完成许多有关图形对象管理的复杂操作和运算。它的真正优势在于对 MapInfo 中的 Table 及其图形对象的管理所提供的特性和强大功能，采用面向对象及事件驱动编程。

(8) 集成二次开发能力

具有 OLE 和 OLE Automation 功能，可以方便地使用 Delphi、VB、VC 等多种开发工具，保证了新的应用程序与原有的应用界面保持一致，为日后的深层开发提供一个扩展空间。

1.1.4 MapInfo 的数据组织

MapInfo 采用双数据库存储模式，即其空间数据与属性数据是分开来存储的。属性数据存储在关系数据库的若干属性表中，而空间数据则以 MapInfo 的自定义格式保存于若干文件中，二者通过一定的索引机制联系起来。为了提高查询和处理效率，MapInfo 采用层次结构对空间数据进行组织，即根据不同的专题将地图分层（图层还可以分成若干图幅），每个图层存储为若干个基本文件。这些文件之间的关系如图 1.2 所示。

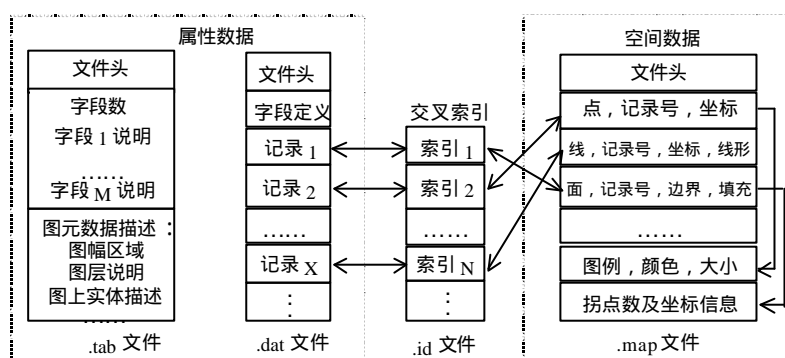


图 1.2 MapInfo 的文件格式及数据关联机制

(1) 属性数据的表结构文件 (.tab)

属性数据表结构文件定义了地图属性数据的表结构，包括字段数、字段名称、字段类型和字段宽度、索引字段及相应图层的一些关键空间信息描述。.tab 文件实际上是一个文本文件，可以在写字板中打开并观察其内容。

(2) 属性数据文件 (.dat)

属性数据文件中存放完整的地图属性数据。在文件头之后，为表结构描述，其后首尾相接地紧跟着各条具体的属性数据记录。

(3) 交叉索引文件 (.id)

交叉索引文件记录了地图中每一个空间对象在空间数据文件 (.map) 中的位置指针。每 4 个字节构成一个指针。指针排列的顺序与属性数据文件中属性数据记录存放的顺序一



致。交叉索引文件实际上是一个空间对象的定位表。

(4) 空间数据文件 (.map)

具体包含了各地图对象的空间数据。空间数据包括空间对象的几何类型、坐标信息和颜色信息等。另外还描述了与该空间对象对应的属性数据记录在属性数据文件中的记录号。这样，当用户从地图上查询某一地图对象时，就能够方便地查到与之相关的属性信息。

(5) 索引文件 (.ind)

索引文件并不是必须的，只有当用户规定了数据库的索引字段后，MapInfo 才会自动产生索引文件。索引文件中每个索引字段都有一个索引表。在每个索引表中，先给出总的数据库记录数目，然后按照索引顺序给出每条属性数据记录在对应的索引字段处的具体属性数据，以及该记录在属性数据文件和交叉索引文件中的记录号。

图 1.3 表示了 MapInfo 的索引文件机制。

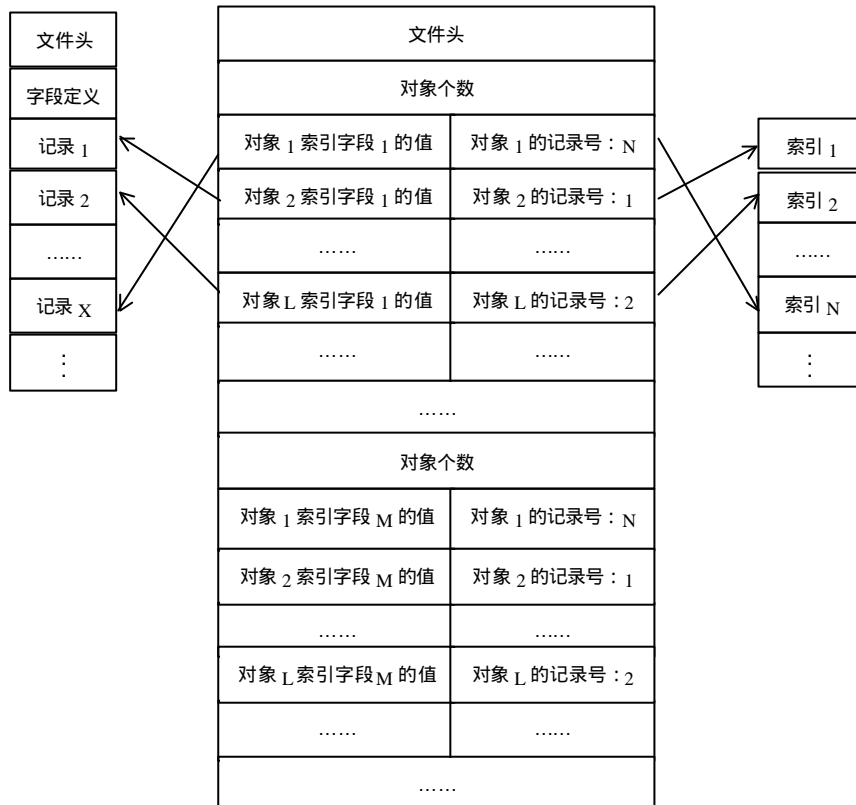


图 1.3 MapInfo 的索引文件格式及数据关联机制

1.2 MapBasic 概述

MapBasic 是在 MapInfo 平台上开发用户定制的应用程序的理想的编程语言。通过使用 MapBasic 进行二次开发，能够扩展 MapInfo 功能，实现程序的自动重复操作并使 MapInfo 与其他应用软件集成。MapBasic 功能强大，用户仅需要几行代码即可在应用软件中实现图

层叠加，并具备其他地理功能。

作为一种功能强大、易于使用的结构化程序设计语言，MapBasic 提供近 400 种函数和命令语句。与传统 GIS 相比，良好的软件集成环境、面向对象和事件处理式的编辑思想，都是 MapBasic 的显著特点。

MapBasic 语言在结构上与 Visual Basic 和 Quick Basic 很相似，基本沿用了 Basic 的控制结构、数据类型与语法规则，但仍有一些区别和特殊之处。当然不能将 MapBasic 和 Visual Basic 相提并论，因为 MapInfo 的特长在于地理信息的应用。

MapBasic 语言具有易学易用的特点，学好该语言的基不出，仍然归结于对该语言语法、变量、表达式、循环和分支、过程和函数等的理解和熟练掌握。

此外，MapBasic 程序易于与用诸如 Visual Basic、Visual C++、C++ Builder、Delphi 和 PowerBuilder 等语言编写的应用软件集成。目前，MapBasic 已被数百个第三方厂商所认可。

1.2.1 MapBasic 的特点

下面是 MapBasic 的主要特点：

(1) MapBasic 是一种类 Basic 语言，可以帮助建立用户界面。通过 MapBasic，用户可以开发个性化的 MapInfo 界面，可以修改或取代标准的 MapInfo 菜单。由于桌面地理信息系统应用软件在各种机构和领域内的应用越来越广泛，用户对开发应用软件的需求日益增大，用户可以利用 MapBasic 开发适合自己需求的应用软件包。

(2) MapBasic 能够实现 MapInfo 与其他应用程序的链接。MapBasic 支持 ODBC 标准、OLE Automation 和 DDE (动态数据交换) 技术，使之易于与其他应用软件相连接。当与其他的商业工具集成时，桌面地理信息系统能够发挥最大作用。例如，利用地图集成技术，能在 Visual Basic、Visual C++、C++ Builder 和 Delphi 等开发工具中集成地图窗口。MapBasic 提供开发的结构，可以调用 Windows 的 DLL 或用 C 或 Pascal 编写的 DLL 等。

(3) MapBasic 能扩展 MapInfo 功能。MapBasic 是一种多平台的、过程化的、事件驱动的、兼容性极强的 MapInfo 应用程序开发工具，其语言结构允许用户定制附加功能。它具有功能很强的内含地理特性的语句和函数。

(4) MapBasic 具有强大的数据库访问工具。使用一条 MapBasic 语句就可以执行复杂而先进的数据库查询。使用功能强大的 MapBasic 语句 (例如 Select、Update)，可以用几行代码代替别的编程语言需要上百行代码才能实现的目标。

(5) 具有较好的兼容性。MapBasic 与所有支持 MapInfo 的平台相兼容，MapBasic 应用程序可以在这些平台上运行。

有了 MapBasic 开发环境的支持，用户可以将 MapInfo 引入各种应用领域 (决策支持、多媒体化、定位系统等)，可以充分展开想象力去开发多姿多彩的地理信息系统。目前，已有上千种使用 MapBasic 开发出的、能够解决商务问题的应用软件。

无论是熟练的还是刚入门的程序员，都能够使用 MapBasic 语言，根据用户的需求开发功能更加强大的桌面地理信息系统应用软件包。



1.2.2 MapBasic 的开发环境

在使用 MapBasic 开发地理信息系统之前，必须先安装 MapBasic。

在安装 MapBasic 之前，必须保证计算机满足以下最低配置要求：任何 80486 或 Pentium CPU；8M 以上内存（若使用 Windows NT，则必须有 16MB 以上内存）；6MB 以上剩余硬盘空间。

在安装 MapBasic 之前，还必须首先安装好 MapInfo 系统。

MapBasic 是一种可编译语言。用户所编辑出的程序可在 MapInfo Professional 或 MapInfo Runtime 上运行。开发环境给用户提供了一个可用于编写程序的文本编辑器。当然，也可以使用原来的文本编辑器。MapBasic 具有编辑和连接模块的功能，可以将程序分模块编写，以减轻工作量。

MapBasic 软件提供了一个应用程序开发环境。在这个开发环境之下，用户可以使用 MapBasic 语言编写应用程序。该集成开发环境如图 1.4 所示，主要包括：

- (1) 文本编辑器 (Editor)：用于生成和编辑 MapBasic 程序的文本编辑窗口。
- (2) 程序编译器 (Compiler)：用于将完成的程序编译成可执行程序。
- (3) 程序链接器 (Linker)：用于将程序的多个模块链接成一个应用程序。
- (4) 联机帮助 (Help)：为用户提供有关 MapBasic 语言及其开发的帮助文档。

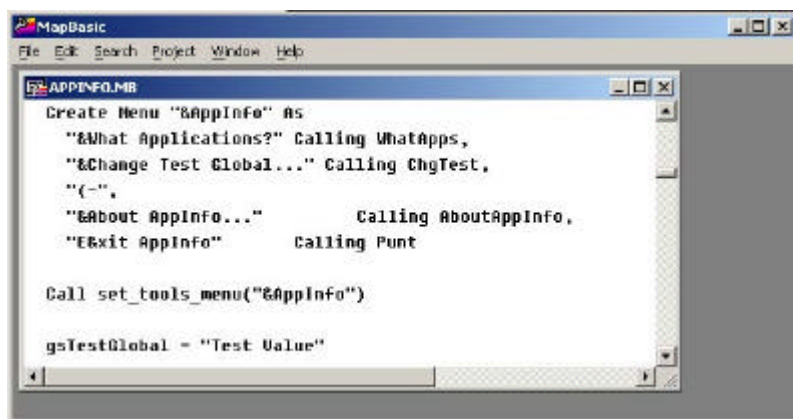


图 1.4 MapBasic 开发环境

针对各类用户的不同需求，用 MapBasic 可以迅速定制出用户所需要的菜单、按钮或对话框等；用户不仅可以修改标准的 MapInfo 菜单，而且可以在原来的菜单栏上增加新的菜单项；利用 MapBasic，用户可以方便、准确地绘制经、纬线，避免手工绘制的繁琐以及可能引起的误差。此外，用户也可以自行设计各种新的图形符号。

MapBasic 中提供了一系列 SQL 函数和许多实用语句，大大增强了对数据库进行操作的能力。此外，MapBasic 还提供了一系列对 MapInfo 的表进行操作的命令和函数，可完成表的各种操作（如打开、新建或修改等），进行对数据对象的各种编辑操作。

利用 MapBasic 还可以将多媒体技术引入地理信息系统，这样可以随时调用视频、声音、动画等信息，使用户的应用系统更加形象、生动。

MapBasic 还提供了 MapInfo 系统与其他应用程序的接口功能，MapBasic 的开放式体系

结构允许程序员调用外部函数中的例程。

1.3 MapBasic 开发过程

每个 MapBasic 程序都将与 MapInfo 同时工作。首先要使用 MapBasic 开发环境生成和编译一个程序，然后运行 MapInfo。一个 MapBasic 程序不是一个独立的程序，它只能在 MapInfo 运行后运行，也就是说，MapBasic 程序在 MapInfo 上运行，所以它可以获取所有 MapInfo 地理数据管理特性的优点。

1.3.1 创建第一个 MapBasic 应用程序

运行 MapBasic 开发环境以后，便可以创建一个 MapBasic 应用程序了。创建一个 MapBasic 应用程序的步骤如下：

- (1) 选择 File 菜单中的 New 命令，打开一个新的编辑窗口。
- (2) 在编辑窗口中输入一些 MapBasic 程序代码。

例如，可以在编辑窗口中输入如下所示的一行代码：

```
Note "你好，世界！"
```

- (3) 选择 File 菜单的 Save 命令，将该程序存储为一个磁盘文件，例如 first.mb。
- (4) 选择 Project 菜单中的 Compile Current File 命令对程序进行编译。MapBasic 编译器首先检查程序语法，如果有语法错误，则显示一个错误信息对话框，并在编辑器窗口的底部显示错误信息。双击任一行错误信息，MapBasic 将滚动到程序的错误处。如果编译成功，将自动生成一个相应的可执行文件，在本实例中为 first.mbx。

注意：当有多个编辑窗口同时打开时，该命令只编译当前活动窗口中的代码。

- (5) 启动 MapInfo。选择 MapInfo 的 Tools 菜单中的 Run MapBasic Program...命令，打开 Run MapBasic Program 对话框。在该对话框中选择编译生成的可执行文件 first.mbx。按“打开”按钮后，MapInfo 将运行该程序。

(6) 或者，在 MapBasic 的集成开发环境中选择 Project 菜单中的 Run 命令，系统将自动启动 MapInfo 系统，并执行刚才生成的可执行文件 first.mbx。

程序运行结果如图 1.5 所示。

在示例中，演示了从创建 MapBasic 程序到编译、运行的全过程。当然，在实际开发过程中，程序比示例程序要复杂得多，但是基本过程是相同的。



图 1.5 程序运行结果

1.3.2 编辑 MapBasic 程序

运用 MapBasic 程序的代码编辑窗口，可以方便地编写 MapBasic 语言的程序代码。由于 MapBasic 是以标准的文本格式来保存程序文件的，所以如果用户愿意，也可以使用其他的文本编辑软件来编辑 MapBasic 程序代码，但是必须以纯文本格式保存文件。



在代码编辑窗口中，可以使用一些快捷键，以提高编辑程序的效率。表 1.1 列出了一些主要的快捷键及其对应的功能。

表1.1 MapBasic代码编辑窗口快捷键及其对应功能

命令	快捷键	功能
File 菜单中的 New 命令	Ctrl + N	打开一个空的编辑窗口
File 菜单中的 Open 命令	Ctrl + O	打开一个 MapBasic 程序文件
File 菜单中的 Save 命令	Ctrl + S	保存当前文件
File 菜单中的 Print 命令	Ctrl + P	打印当前文件
Edit 菜单中的 Undo 命令	Ctrl + Z	撤消当前操作
Edit 菜单中的 Cut 命令	Ctrl + X	将选择的内容剪切到剪贴板
Edit 菜单中的 Copy 命令	Ctrl + C	将选择的内容复制到剪贴板
Edit 菜单中的 Paste 命令	Ctrl + V	粘贴
Edit 菜单中的 Clear 命令	Del	删除选择的内容
Edit 菜单中的 Select All 命令	Ctrl + A	全部选中
Search 菜单中的 Find 命令	Ctrl + F	寻找并替换
Search 菜单中的 Find Again 命令	Ctrl + G	重复最近的寻找命令
Search 菜单的 Replace and Find Again 命令	Ctrl + R	替换选择的文本并继续寻找
Search 菜单中的 Go to Line 命令	Ctrl + T	显示 Go to Line 对话框

MapBasic 的代码编辑窗口所能编辑的文件大小是有限制的。如果用户使用的是 Windows 95 操作系统，那么这个限制是 64KB。换言之，在一个文本编辑窗口中用户能编辑的文件不能大于 64KB。当用户的应用程序比较大的时候，可以采取如下方法之一来处理：

(1) 用其他文本编辑器来编辑 MapBasic 程序文件，再选择 MapBasic 的 File 菜单中的 Compile From File... 来编译该文件。

(2) 把一个比较大的 MapBasic 程序文件分成几个较小的文件，然后使用 MapBasic 的 Include 语句把这些文件组合成为一个应用程序。

(3) 把一个比较大的 MapBasic 程序文件分成几个较小的文件，然后使用一个项目文件 (Project) 把这些文件组合成为一个应用程序。

第三种方法与第二种方法的不同之处在于，第三种方法可以独立编译每个模块。因此如果只修改其中一个模块，只需要对该模块进行重新编译即可。

1.3.3 编译和链接 MapBasic 程序

把程序调入到编辑窗口，或在编辑窗口中编辑新代码，然后选择 Project 菜单中的 Compile Current File 命令，便可以编译当前窗口中的程序代码。

MapBasic 首先检查程序的语法。如果程序中包含语法错误，MapBasic 会先显示一个对话框窗口，告诉用户发现了语法错误。然后在编辑窗口的底部显示错误的详细列表，每个错误信息均由行号开始，它指出了该错误在程序中是第几行。同时 MapBasic 会把错误信息写到一个扩展名为 .err 的文本文件中。只有将程序中所有的语法错误更正以后，MapBasic



才能正确编译程序。

如果用户编辑程序代码时使用的不是 MapBasic，而是其他的文本编辑软件，那么可以选择 File 菜单的 Compile From File... 命令来编译程序。使用该方法不需要在 MapBasic 代码编辑窗口中显示程序。

如果用户选择了 File 菜单的 Compile From File... 命令来编译程序，那么 MapBasic 将首先弹出一个 Compile Program 对话框，要求用户选择一个文件。当用户选择了需要编译的文件并单击“打开”命令按钮后，MapBasic 首先将检查程序中是否有语法错误，如果有错误，MapBasic 只会把错误信息写到一个扩展名为 .err 的文本文件中，而不会在代码编辑窗口中显示错误信息。可以通过选择 File 菜单的 Open 命令来打开该文件，或使用其他的任何文本编辑软件来打开，例如 Windows 自带的记事本、写字板或 Microsoft Word 等。

如果编译成功，MapBasic 会弹出如图 1.6 所示的 Status 对话框。单击 OK 按钮完成整个编译任务，然后就可以运行该程序了。



图 1.6 编译成功消息框

但是 MapBasic 并不能检查到所有的错误。有些类型的拼写错误，MapBasic 编译器是检测不到的。例如，下面程序的第二行代码包含了一个拼写错误——将 Highways 错拼成 Highway：

```
open table "Highways"  
map from Highway
```

MapBasic 编译器并不能识别第二行中的拼写错误，这并不是 MapBasic 编译器的缺陷，因为一些变量和表只有到程序运行时才被计算。当用户运行前面的程序时，MapInfo 试图执行 map from Highway 语句，当然不能正确执行。此时，MapInfo 就会显示一个错误信息“Table Highway is not open”，这说明没有打开 Highway 表。

1.4 创建多个模块的项目

项目文件（Project file）是一个文本文件，它允许 MapBasic 把多个独立的程序文件连接成一个单一的应用程序。如果用户正在开发一个大而复杂的应用程序，程序可能包含上百万行代码。当然，可以把这些代码都输入到一个文件中。但是有经验的程序员是不会这样来管理庞大的程序文件的。因为一旦程序代码超过 1000 行，就很难找到程序中想找的特定部分。因此，许多程序员把一个大的应用程序分成两个或更多个较小的文件，然后建立一个项目文件。由项目文件来通知 MapBasic 链接器如何把那些独立的模块组合成一个可执行的应用程序。

项目文件是 MapBasic 程序设计中的可选择部分。完全可以不使用项目文件，而完成应用程



序的建立、编译和运行工作。然而，如果计划开发一个大型的地理信息系统应用程序，那么使用项目文件会带来一系列的好处：

(1) 项目文件可以模块化程序。一旦建立了一个项目文件，就可以把程序分成许多个小文件。模块化程序的优点是方便维护程序，还可以使每个程序文件变得很小，以便可以在 MapBasic 的代码编辑窗口中进行编辑、编译和修改错误。

(2) 项目文件使多个程序员同时为一个项目编写程序代码变得更加容易。每个程序员可以各自在独立的模块中工作。最后可以用项目文件把这些独立的模块连接起来。

(3) 项目文件可以减少重新编译应用程序的时间。如果多模块的项目中的一个模块被修改，只需要重新编译该模块，然后重新连接这个项目就可以了，这比每次都要重新编译所有源代码要快很多。

1.4.1 创建项目文件

创建项目文件既然有这么多优点，那么如何创建项目文件呢？

下面是一个获取应用程序信息的项目文件 (appinfo.mbp) 的例子：

```
[Link]
Application = appinfo.mbx
Module = appinfo.mbo
Module = ..\Lib\auto_lib.mbo
```

下面是一个创建文本框的项目文件的例子：

```
[LINK]
Application = textbox.mbx
Module = textbox.mbo
Module = ..\Lib\auto_lib.mbo
```

这两个项目文件的最后一行相同，该行代码告诉 MapBasic 把 auto_lib.mbo 模块连接到该项目中。auto_lib 模块是 MapBasic 自带的实例，在 MapBasic 的安装目录中的 SAMPLES\MAPBASIC\LIB 目录中，要成功编译和链接上述项目文件，必须先得到 auto_lib.mbo 文件，默认时，该文件并不存在。需要用户手工将 auto_lib.mb 文件编译成 auto_lib.mbo 文件。如果已经编写了一个程序，并且想为该程序创建一个项目文件，则创建步骤如下：

(1) 选择 File 菜单的 New 命令，打开一个新的编辑窗口。

(2) 在代码编辑窗口中输入关键词[LINK]作为一行，如下所示：

```
[LINK]
```

(3) 在接下来一行输入下面格式的语句：

```
Application = Setting
```

其中 Setting 指的是用户要建立的可执行文件的文件名 (*.mbx) 和路径。例如：

```
Application = d:\MyMapBasicCode\ second.mbx
```



(4) 然后输入格式如下的语句：

```
Module = Setting
```

此时，Setting 代表 MapBasic 的目标文件名，其扩展名为.mbo。当多个模块项目中的一个单独模块被编译时，MapBasic 就会为其创建一个目标文件。例如可以输入：

```
Module = d:\MyMapBasicCode\ Part_1.mbo
```

任何时候，当选择 Project 菜单中的 Compile Current File 时，MapBasic 就会试图把当前文件编译成扩展名为.mbx 的可执行文件。但是如果该程序调用了不在该文件的函数或过程，MapBasic 则不建立一个可执行文件。在这种情况下，MapBasic 假设它只是一个大项目的一部分，因此把它编译成扩展名为.mbo 的目标文件。另外，当一个程序文件没有主过程时（即没有 Main()过程），MapBasic 在编译该文件时为其创建一目标文件。

(5) 重复步骤（4），加入项目文件中需要的其他所有模块。例如可以创建如下所示的项目文件：

```
[LINK]
Application = d:\MyMapBasicCode\ second.mbx
Module = d:\MyMapBasicCode\ Part_1.mbo
Module = d:\MyMapBasicCode\ Part_2.mbo
Module = d:\MyMapBasicCode\ Part_3.mbo
Module = E:\Program Files\MapInfo\MapBasic\SAMPLES\MAPBASIC\LIB\auto_ lib
lib.mbo
```

(6) 选择 File 菜单的 Save As 命令保存项目文件。注意在 Save As 对话框中，选择文件类型中的 Project Files。以便使保存的项目文件的扩展名为.mbp。

(7) 选择 File 菜单的 Close 命令关闭编辑窗口。

这样就创建好了一个项目文件。如果以后要在项目文件中加入更多的模块，只需在该项目文件中加入相应的行（Module =）即可。

1.4.2 编译和链接项目文件

当通过上述步骤创建好项目文件后，便可以编译和链接项目文件，以便生成一个可执行文件。

编译和链接项目文件的步骤如下：

(1) 先编译项目中需要使用的每个模块。为了编译这些模块，可以先选择 File 菜单中的 Open 命令打开模块文件，然后选择 Project 菜单中的 Compile Current File 命令来编译。当然也可以直接选择 File 菜单中的 Compile From File 命令，编译选择的文件。

(2) 显示项目文件。选择 Project 菜单中的 Select Project File 命令，弹出 Select Project File 对话框，在该对话框中选择需要链接的项目文件后，单击 OK 按钮，被选择的项目文件将出现在编辑窗口中。当然也可以通过 File 菜单中的 Open 命令来打开该项目文件。

注意：任何时候只能打开一个项目文件。打开一个项目文件将关闭原打开的项目



文件。

(3) 在 MapBasic 的代码编辑窗口中显示了项目文件后, 选择 Project 菜单的 Link Current File 命令, 链接当前的项目文件。MapBasic 依次读入项目文件中列出的目标文件。如果没有链接错误, MapBasic 将创建一个可执行文件。否则将显示错误信息。

当然, 用户也可以选择 File 菜单中的 Link From File 对项目进行链接, 而不需要先在代码编辑窗口中显示项目文件。但是必须将要链接的各个模块先编译成目标文件。

1.5 调试和运行 MapBasic 程序

创建 MapBasic 应用程序只是应用程序开发工作的一部分, 还需要进行繁重的代码调试工作, 并修改程序中存在的错误, 使之完全按照开发者的意图运行。

1.5.1 错误类型

程序错误可以分为两种类型——编译错误和运行错误。

编译错误主要是由于语法错误或输入错误引起的, 该类型的错误使程序无法成功编译。当编译错误产生时, 相应的错误信息指出了程序文件名和错误所在行号。知道了错误行号, 可以在 MapBasic 开发环境中选择 Search 菜单中的 Go to Line... 命令, 以快速定位存在错误的代码行, 从而方便快速地更正错误。

只要程序编译成功, 就表明程序没有编译错误了。但是, 没有编译错误并不等于程序能够正确运行, 这是因为程序中还可能存在运行错误。运行错误是用户在实际运行应用程序的过程中产生的错误。引起运行错误的原因很多。有可能是算法问题, 也有可能是运行该程序所需要的数据不存在。要纠正这类错误就比较困难了。例如下面的语句能成功编译:

```
Open Table "States"  
Map From States
```

但是如果没有一个名为 States 的表, 那么程序就会产生一个运行错误, MapInfo 中断 MapBasic 应用程序的执行, 然后显示一个描述错误原因的对话框, 如图 1.7 所示。



图 1.7 运行错误的显示

1.5.2 MapBasic 程序的调试

调试程序一般是指发现并修改运行错误的过程。对于运行错误, 有些错误很容易纠正, 例如那些由于程序员输入错误引起的。但更多的是难以定位的错误。

MapBasic 中的 Note 和 Print 语句可以帮助程序员来调试一个程序。例如, 如果想观察一个



变量内容的变化情况，可以简单地在程序中加入如下语句：

```
Print "counter当前的值为：" + counter
```

这样就可以在 MapBasic 的信息窗口中输出一条相应的信息。另外，MapBasic 自带的实例 AppInfo.mbx 可以检查当前正在运行的 MapBasic 应用程序的全局变量值。

为了帮助程序员检测和纠正程序中的错误，MapBasic 还提供了专门的调试工具——Stop 和 Continue 语句。

如果程序运行不正确，可以使用下面的方法找到问题所在：

(1) 在 MapBasic 开发环境中编辑程序，把 Stop 语句放在程序中可能存在错误的那个部分的前面。

(2) 重新编译并运行程序。当程序执行到 Stop 语句时，MapBasic 暂时挂起正在运行的程序，并在 MapBasic 窗口中显示一个调试信息，例如：

```
Breakpoint at second.mb line 23
```

(3) 在 MapBasic 窗口中可以按照下面的方法检查程序：

- 输入“?Dim”查看所有正在使用的局部变量列表。
- 输入“?Global”查看所有正在使用的全局变量列表。
- 输入“?变量名”可以查看某个变量的当前值。
- 输入“?变量名=新值”可以改变某个变量的当前值。

(4) 当检查或修改变量的内容后，可以在 MapBasic 窗口中输入 Continue 继续执行程序，也可以选择 MapInfo 的 File 菜单中的 Continue 来启动程序以重新执行。

在以下情况中，不能使用 Stop 语句来挂起应用程序：

- (1) 在 Function 和 End Function 结构中不能使用 Stop 语句。
- (2) 在对话框控制处理程序中不能使用 Stop 语句。这是因为处理程序处于激活状态，对话框还在屏幕上。
- (3) 在一个进程条 (ProgressBar) 处理程序中不能使用 Stop 语句。
- (4) 当另外一个程序正在运行时不能调试这个程序。

1.5.3 错误的捕获

一个设计良好的应用程序应该对可能出现的运行错误进行预测，一旦错误发生还能采取一些补救措施。对运行错误的窃听和处理就叫做错误的捕获。捕获错误可以使用 OnError 语句。

在程序执行的任何一个点上，错误捕获的执行是有条件限制的。默认情况下，所有过程或函数都是由错误捕获不允许开始的，其中用 OnError 语句可以使错误捕获被允许。典型的情况是，OnError 语句总是指向一个在同一个过程或函数中另一个位置的标号。标号后面的语句叫做错误捕获处理程序。一旦错误发生，就去执行该处理程序，而不是终止程序。

在错误捕获处理程序中，可以调用 Err()函数获取一个整型的代码，它表明了该错误是属于哪一种错误。同样，也可以使用 Error\$()函数返回一个描述错误信息的字符串。在错误



捕获处理程序的结束处使用 Resume 语句，告诉 MapBasic 在该处理程序执行结束时，接着执行哪一程序。

例如下面的代码演示了如何捕获错误并进行处理：

```
OnError Goto no_states
  Open Table "states"
  Map From states

OnError Goto no_cities
  Open Table "cities"
  Map From cities, states

after_mapFrom:
  End Program

no_states:
  Note "不能打开states表"
  Resume after_mapFrom

no_cities:
  Note "不能打开Cities表"
  Map From states
  Resume after_mapFrom
```

该程序代码的第一行语句是使得错误捕获被允许，并告诉 MapBasic，一旦错误发生就转去运行 no_states 后面的处理程序；处理程序的结尾是调用 Resume after_mapFrom，作用是使处理程序结束后能返回到标号 after_mapFrom 处。对于第二个错误捕获语句，与第一句类似。再例如下面一段代码演示了如何在过程中捕获并处理错误：

```
Sub ListSelect
  OnError Goto ERROR_TRAP

  If CommandInfo(CMD_INFO_DLG_DBL) Then
    Call WhatGlobals
  End If

DONE:
  Exit Sub
ERROR_TRAP:
  Note Error$()
  Resume DONE
End Sub
```



第 2 部分 基于 MapObjects 开发 GIS

第 3 章 基于 MapObjects 开发 GIS 概述

MapObjects 是一套制图软件集，它使程序员能够把地图加到应用程序中去。通过 MapObjects 可灵活地建立适合用户的地图接口。在小内存空间中，能用多种工业标准程序环境之一去建立应用程序，能够联合使用 MapObjects 与其他软件去实现地图与用户信息的联系。

MapObjects 是 ESRI 产品的一部分，因此本章首先简单地介绍 ESRI 产品系列，然后介绍 MapObjects 的特点、结构及其数据源，以及如何在 Visual Basic 和 C++ Builder 中使用 MapObject 进行 GIS 的二次开发。

3.1 ESRI 产品系列

ESRI (Environmental Systems Research Institute Inc.) 公司于 1969 年成立于美国加利福尼亚州的 Redlands 市，公司主要从事 GIS 工具软件的开发和 GIS 数据生产。

ESRI 的产品中，最主要的是运行于 UNIX/Windows NT 平台上的 Arc/Info，它主要由两部分组成——Workstation Arc/Info 和 Desktop Arc/Info。其产品还包括 ArcView GIS、MapObjects、ArcFM 和 Internet Map Server。下面分别介绍这些产品。

(1) Workstation Arc/Info

Workstation Arc/Info 基于拓扑数据模型，实现了图库 (Map Library) 的管理，并且具有了栅格数据的分析功能，支持栅格矢量一体化查询和叠加显示。此外，Arc/Info 还提供了二次开发语言 AML 以及开放开发环境 ODE，以便于用户定制自己的 GIS 应用。

Workstation Arc/Info 提供了最基本的 GIS 功能，包括数据录入和编辑、投影变换、制图输出、查询分析以及分析功能 (缓冲区分析、叠加复合分析等等)。

除了上述基本功能以外，Workstation Arc/Info 还通过一些扩展模块实现特定的专门功能：

- TIN——基于不规则三角网的地表模型生成、显示和分析模块，可以根据等高线、高程点、地形线生成 DEM 数字高程模型，并进行剖面、填挖方计算等。
- GRID——栅格分析处理模块，可以对栅格数据进行输入、编辑、显示、分析、输出，其分析模型包括基于栅格的市场分析、走廊分析、扩散模型等。
- NETWORK——网络分析模块，提供了最短路径选择、资源分配、辖区规划、网络