第 1 章　MapBasic 基础

MapBasic 语言作为 MapInfo 地理信息系统的开发工具，是美国 MapInfo 公司于 1985 年推出的，最初版本为 1.0，其后随着 MapInfo 地理信息系统版本不断更新，MapBasic 语言的版本也随之更新，其功能也不断得到增强，到 1996 年 MapInfo 公司推出 MapBasic 语言的 4.0 版本。作为 MapInfo集成开发工具的 MapBasic 4.0，它具有强大的地理信息操作功能、丰富的程序语句和完善的各类函数。

本章将以 Mapbasic 4.0 版本为基础，详细分析和介绍 Mapbasic 语言的基础知识。

MapBasic 能够让用户自如地应用 MapInfo 桌面地理信息系统。MapBasic 提供一个集成开发环境，使用该集成开发环境，用户能完成 MapBasic 程序的编辑、编译、调试和运行，并提供良好的在线帮助。

MapBasic 集成开发环境由以下几部分组成：

- 文本编辑器，用于输入和编辑 MapBasic 应用程序。
- MapBasic 编译器，当用户编辑完一个应用程序后，可以将它编译成可执行的（“executable”）应用程序（即能被 MapInfo运行的应用程序）。
- MapBasic 链接器，用户若建立了一个大型的应用程序，可以把程序分成几个部分，然后，将各个部分链接（“link”）在一起成为一个完整的应用程序。
- MapBasic在线帮助，提供 MapBasic语句和函数的详细说明，语法参考信息及例程的注释和说明。

MapBasic 语言与 Microsoft 的 QB 和 VB 有许多相似之处，但是，它的独特之处是其地理信息操作功能。使用 MapBasic 语言可以使用户方便地开发专门的 MapInfo 应用系统。

传统 BASIC 编码范例	MapBasic 编码范例
20 GOSUB 300	Call check_Status(quit_time)
30 IF DONE=1 Then GOTO 90	Do while Not quit_time
40 FOR X=1 TO 10	For x=1 to 10
50 GOSUB 4000	call Process_batch(x)
60 NEXT X	Next
80 GOTO 30	Loop

用 MapBasic 集成开发环境创建和编译的程序，必须在 MapInfo 环境下运行。当运行 MapBasic 程序时，系统自动启动 MapInfo。MapBasic 程序在 MapInfo 环境下运行后，它就具有了所有 MapInfo 的地理数据操作管理能力。

1.1　软、硬件配置要求

在安装 Windows环境下的 MapBasic 之前，你的计算机必须具有下述最低配置要求：

硬件/软件	配置
操件系统	Microsoft Windows 3.1 或更高版本 Micocsoft Windows NT3.51 或更高版本 或 Micosoft Windows 95/98
主机	80486 或奔腾及以上计算机
内存	8MRAM(使用 WindowsNT 需 16MRAM
显示	Windows 支持的各种适配器及显示器
鼠标	Windows 兼容的各种鼠标
硬盘空间	完全安装：6M 剩余空间 最小安装：2.5M 剩余空间

1.2 MapBasic 安装和运行

1. 先行安装 MapInfo4.0；

2. 运行 SETUP.EXE，然后按 OK 按钮；

3. 根据提示执行后面的操作。注意在工作站配置（Work station configration ）中，一般选择 standalone workstation。

运行 MapBasic 应用程序或双击 MapBasic 图标，即可运行 MapBasic 。注意：MapInfo Professional 能运行 MapBasic 早期版本的应用程序，MapInfo 3.0 不能运行用 MapBasic 4.0 建立的应用程序。

1.3 文件名和文件类型

安装 MapBasic 后，在其目录下包含以下文件：

errors.doc ：错误信息文件，显示 MapBasic 错误代码

mapbasic.exe ：运行 MapBasic 系统的可执行文件

mapbasic.def:：组件代码定义文件

menu.def ：菜单代码定义文件

icons.def：按钮、用户描述组件文件

mapbasic.hlp:：MapBasic 在线帮助文件

mapbasic.h：C/C++语法头文件；其内容类似于 mapbasic.def，但是使用的是 C/C+语法规则。

mapbasic.bas：Visual Basic 语法文件；其内容类似于 MapBasic.def，但是使用的是 Visual Basic 语法规则

.mb、.mbp：程序范例

用户输入的 MapBasic 源程序保存为一文本文件，扩展名是.MB 文件。

应用程序是由 MapBasic 生成的二进制文件。在编译程序时，MapBasic 将创建应用程序，MapBasic 应用程序文件的扩展名是.MBX(MapBasic 文件扩展名）

使用 MapBasic 系统创建应用程序时，会生成如下几类文件：

mb：程序文件（源代码）、 .mbx：编译文件、.mbp：工程文件（显示该工程中所有项目）、.mbo：对象文件（编译后该工程各项所创建的文件） 、.err：若在编译时存在错误，将记录这次错误。

1.4 建立和运行 MapBasic 应用程序

用户可以按照以下步骤创建一个简单的 MapBasic 程序：

1. 运行 MapBasic 集成开发环境；
2. 选择菜单 File 下 New 命令打开一个编辑窗；
3. 在编辑窗中，输入 MapBasic 程序，例可以输入如下命令：

   Note “Welcome to MapBasic!”
4. 选择菜单 File 下 Save 命令存盘且输入文件名，如输入文件名：

   Welcome.mb.

   注意，不要关闭编辑窗；
5. 选择菜单 Project 下 Compile Current File 命令，编译后生成一个可执行的应用程序文件(Welcome.mbx)；
6. 选择 Project 下 Run 命令，执行建立的程序；也可以先运行 MapInfo，选 择 MapInfo 菜单 File 下的 Run 命令。MapInfo 提示你选择要运行的程序，选择 Welcome.mbx，MapInfo 将运行该程序；
7. 程序运行的结果是：在一对话框中显示信息“Welcome to MapBasic!”。

上述是创建、编译、运行 MapBasic 应用程序的主要步骤。当然，在实践中过程要更复杂，例如，这个过程没有描述遇到意外错误的情况。创建、编译 MapBasic 程序更多的内容，请见第 3 章“使用集成开发环境”。

5 MapBasic 的特点

1. 方便地定制 MapInfo

通过 MapBasic ，用户能定制 MapInfo 用户界面。MapBasic 应用程序能够修改或取代标准的 MapInfo 菜单，能增加全新的菜单项到 MapInfo 菜单栏中。

2. MapBasic 能使 MapInfo 自动操作

MapBasic 应用程序能够减轻 MapInfo 用户手工操作的烦恼。例如，MapInfo 用户开发一个地图，当需要绘制经纬度构成的格子线时，如果手工绘制是非常繁琐的，而采用 MapBasic 编程的方法，不仅能提高绘制的精度，而且能提高效率。

3. 强大的数据存取功能

通过 MapBasic 语句，能够完成复杂、多样的数据查询。通过 MapBasic 的 Select 语句（它与结构化查询语句 SQL 中的 Select 语句类似），能查询、过滤、排序、统计数据。

使用像 Select 和 Update 这样的 MapBasic 语句，只需几行代码就可以完成其他语言十几行甚至上百行语句才能完成的工作。

4. 可移植性

MapBasic 应用程序是可以移植的。在 Windows 环境下开发的 MapBasic 程序，也可以在 Macintosh 环境下的 MapInfo 上运行。

如果开发 MapBasic 应用程序时，采用了 Microsft Windows 下才有效的 DDE 这样的技术，那么对操作平台就有依赖性了。不过只要在开发应用程序时，稍稍考虑一下这类问题，程序还是非常容易移植的。

5. 与其他应用程序链接

MapBasic 提供一个开放的结构，在 MapBasic 程序中可以调用外部库。 MapBasic 语言能使用 Dynamic Data Exchange(DDE）与其他软件包进行链接，包括 Visual Basic 应用程序；MapBasic 语言也能链接 Windows 下 Dynamic Link Library (DLL) 文件。用户可以从商业性软件包中获得 DLL 文件，也可以使用像 C 或者 Pascal 这样的编程语言建立你自己的 DLL 文件。同样，Macintosh环境下建立的 MapBasic 程序能够使用 Apple Events，XCMP 和 XFCN，以扩展 MapBasic 语言。

MapBasic4.0 提供了一个新功能 ——Integrated Mapping（集成地图），可以集成 MapInfo 功能到其他环境下开发的应用系统中，像在 Visual Basic、Delphi 等开发环境下可以集成 MapInfo 的功能，使开发的应用系统功能更强大。

1.6 MapBasic在线帮助

MapBasic 集成开发环境提供了丰富的在线帮助。在线帮助提供了语言中的每一语句和函数的描述，同时也提供一些使用 MapBasic 集成开发环境方面的帮助信息。

输入程序时，如果选择了一语句或函数，然后按 F1，帮助窗口将显示该语句或者函数的帮助信息，帮助系统中还有一些简单的范例，你能从帮助窗口中复制它，并粘贴到你的程序中。

若正处在帮助状态下，点击 MapBasic 窗口，帮助窗口将隐藏起来。但是这个帮助窗口并没有被关闭，只是它处于非活动状态。按 ALT-Tab 键能重新激活帮助窗口。

1.7 MapInfo 的 MapBasic 交互命令窗口

MapInfo 系统提供了一个学习 MapBasic 语言的方法。这个窗口可以帮助用户学习 MapBasic 语言的使用。

1. 运行 MapInfo

2. 选择菜单 Options 下 Show MapBasic Window 命令

MapBasic 窗口出现在屏幕上后，使用 MapInfo 菜单和对话框时，MapBasic 窗口会显示相应的 MapBasic 语句。如果使用 MapInfo 的 Select 对话框完成了一项查询，MapBasic 窗口会自动显示完成同样功能的 MapBasic 语句。

相反，可以直接在 MapBasic 窗口中输入 MapBasic 语句，完成对 MapInfo 的操作，注意不是所有的 MapBasic 语句都能这样使用。能否这样使用，在线帮助中有明确的说明。

第 2 章　MapBasic 集成开发环境

2.1　集成开发环境简介

MapBasic 集成开发环境包括文本编辑器、编译器和在线帮助。下拉菜单——File、Edit、Search、Project、Window 和 Help，能够帮助用户创建和编辑程序、编译程序、运行应用程序以及使用在线帮助。

MapBasic 的文本编辑器使用非常方便。文本编辑器具有一般编辑器中常用的功能。File 菜单包括 Open、Close、Print 和 Save 等命令；Edit 菜单有 Undo、Cut、Copy 和 Paste 等命令。

编译器将源程序编译为可执行程序，其文件扩展名为 .MBX 或 .MBO。

在线帮助系统提供了 MapBasic 全部程序语句和函数的功能、语法的详细说明，以及简单例程的详细说明，可以为用户提供良好的在线帮助。

2.2　编辑 MapBasic 应用程序

运行 MapBasic，在菜单 File 下选择 Open 或 New。在编辑窗口中就可以输入程序行了，假定是一个新程序输入编辑窗口，可以输入以下一行 MapBasic 程序：

```
Note "Welcome to MapBasic! "
```

输入完成程序后，在菜单 File 下选择 Save 命令把程序存盘，输入文件名时，只需要输入程序主文件名，MapBasic 会自动追加扩展名 .mb 给程序文件。假如程序取名为 Welcome，实际存储的文件名是 Welcome.mb。

MapBasic 源程序是以纯文本形式保存的，所以也可以采用其他的文本编辑软件编辑源程序。为便于使用 MapBasic 集成开发环境，下表列出了在 MapBasic 编辑窗口中可以使用的快捷键和鼠标热键：

组　　合键	功　　能
Home/End	光标移到行首／行末
Ctrl-Home/Ctrl-End	光标移到文本首／末
Ctrl-←/Ctrl-→	向前／向后移动一个词
Ctrl-T	显示定位行的对话框
Ctrl-O	显示 Open（打开文件）对话框
Ctrl-N	打开一个新的空白编辑窗口
Ctrl-S	保存当前编辑的文件
Ctrl-P	打印当前编辑窗口文件

Ctrl-A	选择编辑窗口下所有文本
Ctrl-C	复制所选文本到剪贴板
Ctrl-X	剪切所选文本到剪贴板
Ctrl-V	将剪贴板文本复制到编辑窗口
Ctrl-Del	删除光标后的内容
Del	删除所选文本，但不复制到剪贴板
Ctrl-F	显示查找和替换对话框
Ctrl-G	重复查找命令
Ctrl-R	替换所选择的文本，开始下一次查找
Ctrl-J	显示选择工程文件对话框
Ctrl-K	编译当前窗口的程序
Ctrl-E	移动编辑窗口中产生编译错误的行
Ctrl-L	链接当前工程文件
Ctrl-U	运行当前应用程序
F1	显示帮助；若选择了一个函数名，显示该函数的帮助内容
F8	显示文本设置对话框
Ctrl-F4	关闭当前编辑窗口
Alt-F4	退出 MapBasic 集成开发环境
Shift-F4	平铺窗口
Shift-F5	层叠窗口

鼠标热键及作用

鼠标动作	作　用
双击(Double-click)	在编辑窗口中，选择程序的一个词；在错误信息窗口中，定位到产生错误的行
连击 3 次(Triple-click)	选择整行
拖放(Drag & Drop)	拖动文本到另一窗口；在同一窗口中移动文本，如果拖动时按下了 Ctrl 键，则复制该文本

MapBasic 帮助中包含了很多例程，可以拖放这些代码到编辑窗口中。

注意，MapBasic 的文本编辑器也有一些限制。每个 MapBasic 编辑窗口对编辑文本大小有一定的限制。在 Window 3.1 中，其长度最大为 50K；在 Window95 中，最大为 64K。当在编辑窗口插入文本时产生蜂鸣，就表示编辑窗口的文本长度达到了极限。

有三种方法能够避开文本大小限制：

1. 采用其他文本编辑软件编写程序，编译程序时，在 MapBasic 下，用 File 菜单下的 Compile From File（编译）命令即可。

2. 把程序（.mb 文件）分成 2 个或多个小文件，然后使用 MapBasic 的 Include 语句，把这些小文件链接成单一的应用程序。

3. 把程序（.mb 文件）分成 2 个或多个小文件，然后创建 MapBasic 工程文件，把这些小程序链接成单一的应用程序。用这种方法，与使用 Include 语句相似。不过采用工程文件

的方法更方便，因为工程中的每一个文件能独立编译生成.MBO 文件，而修改时只需对修改过的小程序进行编译，然后再将多个.MBO 文件链接起来生成一个应用程序即可。

2.3 编译和运行 MapBasic 应用程序

如果在 MapBasic 编辑窗口下已经编写了一段程序，就可对程序进行编译（Compile）。选择 project 下的 Compile Current File 命令，即可对程序进行编译。当打开了多个编辑窗口时，编译器只对 MapBasic 当前窗口的程序进行编译。

MapBasic 编译器会检查程序的语法。若程序有语法错误，就显示找到错误的对话框图，然后在编辑窗口下面的信息窗口中显示产生错误的程序行及错误说明。只有排除了所有错误，才能编译成功。为快速找到产生错误的程序行，用鼠标双击错误信息行， MapBasic 会立刻定位到产生错误的行。更正程序错误以后，再次选择 Compile Current File 命令重新编译，一旦编译成功，MapBasic 显示一编译完成的对话框。编译成功后，MapBasic 创建一个 .mbx 文件。

MapBasic 编译成功并不表示源程序中没有错误。有些拼写错误，编译器不能检测出来。例如：MapBasic 会编译通过如下程序，尽管第二行输入有错误（table name 写成 tbale_name）

```
Open Table "table_name"
Map From tbale_name
```

MapBasic 编译器不能识别第二行的拼写错误，它不能被编译器检测到，当程序运行时，MapInfo 会试着执行 Map From tbale_name 语句，这时 MapInfo 会提示不能打开表的错误信息。

注意，采用其他熟悉的编辑器，也能编辑 MapBasic 程序，但一定要以纯文本文件格式保存 MapBasic 程序。当采用这种方法编辑程序时，编译时使用 MapBasic 下的 File>Compile From File 命令编译程序。如果编译过程中有错误，将生成一个扩展名为 .err 的文件，要察看错误，选择 File>Open 命令打开这个文件。

要运行编译好的程序，可从 MapBasic 集成开发环境中快捷地运行程序，选择 Project>Run（或者按 Ctrl-Ù 命令，执行当前编辑的程序；在 MapInfo 下要执行一个应用程序，选择 File>Run MapBasic Program 命令，MapInfo 将提示选择要运行的 MapBasic 应用程序（.mbx）。

2.4 MapBasic 工 程文件

2.4.1 什么是工程文件

工程文件是一个文本文件，它将几个独立的程序链接成一个应用程序。如果开发大型复杂的应用程序，为便于管理、编辑程序，以及为避开 MapBasic 文本编辑器的限制，需要将大型程序分成 2 个或多个小文件。如果分成多个文件，就必须建立一个工程文件，工程文件会告诉 MapBasic 链接器，如何把若干独立的程序链接成单一的可执行应用程序。工程文件是可选的，不使用工程文件也能创建、编译和运行应用程序。不过，如果计划开发大型的 MapBasic 应用程序，就需要运用 MapBasic 工程文件的优越性了。使用工程文件有以下三个优点：

1. 工程文件使调试程序更容易，一旦创建了一个工程文件，就能将程序分成多个小程序文件，小型程序通常易于维护，而且在 MapBasic 编辑窗口中，程序太大也不易编辑修改。

2. 工程文件可以让几个程序员同时工作，一旦建立了一个工程文件，每一个程序员都能使用各个独立的程序文件，以后再将各个独立的程序通过工程文件链接起来。这样能够提高应用程序的开发效率。

3. 工程文件能够缩短重新编译应用程序的时间，如果改变了有多个文件的工程文件中的一个文件，只须编译这个文件，然后重新链接到工程中，这比不使用工程文件而重新编译所有的程序代码快得多。一个典型的工程文件如下所示，该工程文件的文件名是 SAMPLE.mbp：

```
[Link]
Application = SAMPLE.MBX
Module = comm.mbo
Module = data_h1.mbo
Module = data_q1.mbo
```

MapBasic 软件包中有许多例程可供编程使用，如果在工程文件中有类似 Module=auto_lib.mbo 这样的行，在链接时，工程文件告诉 MapBasic 建立 auto lib 单元到工程中。至于每一个例程的具体功能及用法参见相应的文档说明。

2.4.2 如何创建工程文件

创建工程文件可采用以下步骤：

1. 选择 File 菜单下 New 命令打开一个新的编辑窗口。
2. 在编辑窗口输入下列命令：

   [Link]
3. 输入 Application=appfilename (appfilename 是将要创建的可执行文件的文件名），如：

   Application=C:\MB\ SAMPLE.MBX
4. 输入 Module=modulename(modulename 是 MapBasic 各单元的文件名），如：

   Module=C:\MB\CODE\ data_h1.mbo

不管什么时候选择 Project 下 Compile Current File 命令，MapBasic 都会尽力将输入的文件编译成可执行的应用程序（扩展名 .mbx)。但是如果程序文件涉及到的函数或过程不在这个文件里，MapBasic 就不能创建 .mbx 文件，而是建立单元文件（.mbo)取代可执行文件（.mbx)。无论什么时候，即使是没有主程序，MapBasic 也能创建单元文件。

5. 重复第 4 步的操作，将要包含到应用程序中的每一文件添加到工程文件中。
6. 选择 File 菜单下 Save As 命令保存工程文件，在 Save As 对话框中，选择文件类型为“Project File”(从对话框左下角的文件类型表中选择),这样文件的扩展名是 .mbp(MapBasic Project)。
7. 关闭编辑窗口。

注意，如果以后想添加更多的单元到工程文件中，只须在工程文件适当位置添加“Module= ”行即可。

2.4.3 编译、链接工程文件

建立好工程文件后，就能通过如下步骤编译和链接工程文件：

1. 编译工程文件的每一个单元，选择 File 菜单 Open 命令打开一个单元文件，然后选择 Project 菜单下 Compile Current File；若编译一个没有打开的单元，可选择 File 菜单下 Compile From File.

2. 选择 Project 下 Select Project File,告诉 MapBasic 想要链接的工程文件。在 Select Project File 对话框中选择工程文件（.mbp），然后按 OK，所选的工程文件就调到一编辑窗口中，这个文件一直都打开，直到退出 MapBasic，并且每次只能打开一个工程文件。不能通过 File 下 Open 命令打开工程文件，要打开工程文件，必须用 Project>Select Project File 命令。

3. 选择 Project 下 Link Current Project 链接应用程序。MapBasic 在工程文件中读取单元文件(.mbo)表,若没有链接错误，MapBasic 就建立一可执行文件 (.mbx)。若有错误，MapBasic 会显示错误信息。若工程文件不是当前窗口，也可以通过 File 下 Link From File 命令完成链接。

用 MapBasic 编译器创建的单元文件不能用其他的链接器链接（如： C 语言的链接器），只有 MapBasic 链接器才能链接 MapBasic 单元文件。

2.4.4 多个程序编程

如果使用工程文件，有时需要同时打开工程中的所有文件，打开程序对话框允许同时打开多个文件。只需在单击另一文件名时按下 Shift 或 Ctrl 键。按下 Shift 键选择两文件间的所有文件，按下 Ctrl 键，选择多个不连续的文件。

如果一个 .MB 文件是多单元工程中的一部分，它可以使用其他单元的函数和过程。如：textbox.mb 调用 HandleInstallation 过程，而该过程在 auto_lib 单元中。

MapBasic 程序调用外部过程，在程序中必须用 Declare Sub 语句对子程序进行声明；同样，如果程序调用外部函数，程序中也必须用 Declare Function 语句对函数进行声明。这些 Declare 语句会告诉 MapBasic 编译器，这些过程或函数使用哪些参数。

2.4.5 全局变量与局部变量

声明全局变量有两种方法。一是直接在程序中用 Global 语句进行声明；另外，也可以通过定义文件的方式声明。采用定义文件方式的方法是：

1. 建立一个定义文件（如“ globals.def” ）并在这个文件中定义全局变量。

2. 用 Include 语句将全局变量添加到需要使用的程序中。

如：auto lib.def 文件声明了 2 个字符串全局变量 gsAppFilename 和 gsAppDescription。如果多个程序需要使用这两个全局变量，只要在程序开头部分加入如下语句：

```
Include "auto lib.def"
```

这样，在一个单元中改变全局变量的值时，另一个单元中该变量的值也一同改变。

全局变量也可用来与其他应用系统共享信息，可参见有关 Dynamic Data Exchange（DDE）方面的介绍。

局部变量只能在一个函数或过程中使用，声明局部变量用 Dim 语句。

2.5 集成开发环境菜单简介

2.5.1 文件菜单（File）

文件菜单（File）提供建立、打开、关闭、存盘、退出、打印等命令。

1. 新建（New）打开一新的编辑窗口，可以在窗口中输入程序。

2. 打开（Open） 在新窗口中打开一个文件，这个文件可以是 MapBasic 源程序文件(.mb)、错误信息文件（.err）或 MapInfo 工作区文件（.wor）。

3. 关闭（Close） 关闭活动编辑窗口，如果已改变窗口的内容， MapBasic 将提示是否存盘保存所做的修改。

4. 全关闭（Close All） 关闭所有打开的编辑窗口。和 Close 命令一样，MapBasic 也会提示是否存盘保存所做的修改。

5. 保存（Save）保存活动编辑窗口的内容到磁盘上。只有改变了活动编辑窗口的内容 Save命令才有效。

6. 另存为（Save As）用一个新的文件名保存活动编辑窗口的内容。

7. 恢复（Revert）放弃上次存盘后所作的修改，只有已改变了活动编辑窗口的内容 Revert命令才有效。

8. 编译文件（Compile From File） 编译已存在的.mb 磁盘文件，而不是编译当前窗口中的程序。如果有编译错误，编译器将错误信息写入名为 filename.err 的文本文件，要查看该文件内容，选择 File 下 Open 命令即可。

9. 链接文件（Link From File） 链接一个不在当前编辑窗口的工程文件。

10. 打印设置（Page Setup）定义打印机各个选项。

11. 打印（Print）打印当前编辑窗口的内容。

12. 退出（Exit） 退出 MapBasic 集成开发环境。MapBasic 会提示存盘或者放弃所做的修改。

2.5.2 编辑菜单（Edit）

Edit菜单提供编辑修改 MapBasic 程序所使用的编辑命令。

1. 撤销（Undo） 撤销最后一次在编辑窗口所作的修改。当执行 Undo命令后，MapBasic 放弃最后一次的修改，并变成 Redo;若选择 Redo，MapBasic 又会重新恢复所做的修改。

2. 剪切（Cut）剪切所选的文本（高亮显示）到剪贴板。

3. 复制（Copy） 复制所选的文本到剪贴板。

4. 粘贴（Paste）拷贝剪贴板的内容到当前编辑窗口的光标位置，若在窗口中选择了一块文本，使用 Paste命令，剪贴板的内容将替换掉所选的文本的内容。

5. 删除（Clear）删除所选的内容，但不复制到剪贴板。

6. 全选择（Select All）选择当前编辑窗口中的全部内容。

2.5.3 查找菜单（Search）

查找、定位和替换编辑窗口的文本内容，这些命令简化了查找语法错误的过程。

1. 查找（Find） 在当前编辑窗口查找指定的字符串或文本。

2. 再查找（Find Again） 重复查找上次 Find 命令所查找的内容。

3. 查找并替换（Replace And Find Again） 用指定的字符串替换上次 Find 对话框所指定的内容，然后继续查找并用高亮度显示匹配内容。

4. Next Error 当程序不能正确编译时，MapBasic 在编辑窗口的下部会显示错误列表。Next Error 显示编辑窗口中程序下一个有错误的行。

5. Previous Error 与 Next Error 相似，previous Error 显示的是错误列表中前一错误的程序行。

6. Go To Line 提示你输入行号，然后在编辑窗口显示的程序中定位到该行。

程序在编译成功后，运行时仍可能有错误，这时出现一对话框，显示程序中有错误的行。一般来说，返回到 MapBasic 集成开发环境后就会定位到程序出现错误的行。

2.5.4 工程菜单（Project）

工程菜单用来编译和运行 MapBasic 程序，显示程序结果以及显示或隐藏错误信息窗口。

1. Select Project File 打开一个已有的工程文件，一旦打开了一工程文件，该工程文件就成为当前工程文件，且能通过 Link Current Project 命令链接生成一个应用程序。

2. Compile Current File 编译当前编辑窗口中的程序。

若编译器在程序中检测到语法错误， MapBasic 会在编辑窗口的下部显示错误列表，若没有错误，MapBasic 就建立 Mbx 文件（如果该单元是独立的程序的话）或工程单元（mbo）文件。

3. Link Current Project 链接当前工程文件。

4. Run 执行当前编辑窗口的应用程序。

5. GetInfo 显示当前编辑窗口程序的有关信息，如盘符、路径、修改时间等。

6. Show/Hide Error List 显示或隐藏当前编辑窗口的错误信息列表窗口。如果有错误信息窗口，则菜单项为 Hide Error List ；如果隐藏了错误信息窗口，则菜单项为 Show Error List。

2.5.5 窗口菜单（Window）

若打开了多个编辑窗口， Window 菜单允许排列这些窗口或者切换窗口的活动状态。

1. Tile Windows 平铺编辑窗口。

2. Cascade Windows 层叠编辑窗口。

3. Arrange Icons（重排图标）用最小化窗口的图标组织排列窗口。

4. Text Style 改变窗口显示。

5. Window 菜单下部会显示打开的每一个程序，要激活其中的一个窗口，从 Window 菜单下选择该项即可。

2.5.6 帮助菜单（Help）

使用 Help 菜单可使用系统帮助，帮助文件描述了 MapBasic 语言的所有语句和函数。一些帮助中包括简单的程序范例，可以把这些程序复制到剪贴板，然后粘贴到自己的程序中。如果使用的是 Windows95/98，则可以从帮助窗口中拖出这些内容，放入自己的程序中。

第 3 章　MapBasic 编程基础

本章详细介绍 MapBasic 语言的语法涉及的基本内容，以及 MapBasic 应用程序的结构和组织方法。包括 MapBasic 语言语法的一般规则、变量、数据类型、数组、表达式、运算符、循环控制、条件分支控制及其他流程控制命令；还包括 MapBasic 应用程序的过程、系统保留过程、函数及自定义函数、以及应用程序的编辑和调试方法等方面的内容。

3.1　MapBasic 语法

本节介绍 MapBasic 语言的语法，主要是讨论以 MapBasic 应用程序结构组织为主的内容，关于 MapBasic语言每条语句和每个函数的语法介绍，请参见本书的第 10、11 章。

3.1.1　一般语法规则

MapBasic 语言语法的一般规则主要包括以下几个方面：

1. 注释

MapBasic 和其他的 Basic 语言一样，用单引号作为注释符的开始。当程序中出现“ ' ”时，MapBasic 将其后面的内容当作注释处理，除非“ ' ”，包含在一定字符串中间。

2. 字母大小写

MapBasic 编译器不区分字母大小写，可以用大写、小写或大小写字母混合编程。在 MapBasic 编程时，建议用户将每一命令关键字的第一个字母大小，其余为小写。

3. 跨行输入

在编写 MapBasic 程序时，可以跨行输入。例如下面的程序代码就是跨行输入 If...Then 语句。

```
If counter = 55
        Or counter = 34 Then
  Note"Counter is invalid"
End If
```

4. 嵌入 MapBasic.def

Mapbasic.def文件是一文本文件，它定义了一些标准的 MapBasic代码。通常，Mapbasic.def 中的代码都是大写字母组成的变量名（如 TRUE、FALSE、BLACK、WHITE、CMD INFO X、OBJ_INFO_TYPE 等等）。在 MapBasic 提供的例程中，经常能看到这些代码，如：

```
If  CommandInfo( CMD_INFO_DLG_OK  ) Then
```

如果用户在程序使用了这些标准代码（如：上例中 CMD_INFO_DLG_OK），则必须在程序中编写类似 Include "mapbasic.def" 这样的语句，以包含 MapBasic.def，否则，在运行时程序会出现错误（例如：“Variable or Field CMD_INFO_DLG_OK not defined”）。这就是所谓“ 嵌入标准代码文件 ”。

5. MapInfo 交互命令窗口

MapInfo 系统中，可以启动一个 MapBasic 交互命令窗口，在该窗口中可以直接输入和执行 MapBasic 命令语句，帮助学习语法和应用程序结构，还可以用来调试 MapBasic 应用程序。不过有许多 MapBasic 语句并不能直接在 MapBasic 交互命令窗口中使用。

MapBasic 命令语句能否在 MapInfo 的 MapBasic 窗口中执行，在 MapBasic 参考手册或者在线帮助中有明确的说明。一般来讲，组合命令语句（如 If...Then、For...Next 和 Go To 等)不能输入到 MapBasic 窗口。

如果直接输入语句到 MapInfo 的 MapBasic 窗口的时候，需要跨行输入，按 Ctrl-Enter，而不能按 Enter。输入完整个语句，高亮显示这些行后按 Enter。

在 MapInfo 的 MapBasic 窗口不能使用 MapBasic.def 中定义的代码，而只能输入具体的值。每一个代码都有一个对应值，如代码 BLACK 的值是 0。具体对应的值可以打开代码定义文件查阅。

3.1.2 变量、数据类型和数组

MapBasic 语言的变量声明和变量赋值与其他现代 BASIC 语言极为相似。不过，MapBasic 支持的某些变量类型其他语言并不支持，典型的是 Object 型变量。

1. 变量声明及赋值

所谓变量其实就是计算机中一块非常小的内存区域。在编程过程中，要临时保存各类信息，就需要声明一些变量。每一个变量都有唯一的名字。每声明一个变量， MapBasic 都会开辟一小块内存区域来保存信息。

用 Dim 语句声明局部变量。每一个变量必须在使用之前声明。

使用“=”（等于号）给变量赋值。变量声明及赋值的方法如下：

Dim 语句不仅能声明一个或多个同类型的变量，也能声明多个不同类型的变量，变量之间用逗号“，”分隔开，下面是几个声明和赋值的例子。

```
Dim Counter As Integer
Dim  x，y，langitude，latitude As Float
Dim start_date，end_date As Date，  First_name，last_name As String
counter=10
x=12.01
y=-100.001
start_date="9/9/99"
```

2. 变量命名规则

变量名必须符合下述规则：

每个变量名最长不能超过 31 个字符；变量名中不能有空格；变量名必须以字母、“.”、“~”开头；变量名由字母、数字、“#”、“_”组成。变量名可以用下列字符结尾：$、%、&、或 @。在一些 BASIC 语言中，这些字符用于指定变量类型；但在 MapBasic 中它们没有特殊的含义。

不能使用 MapBasic 的关键字作为变量名。

3. 数据类型

MapBasic支持下列变量类型：

类　型	意　义
SmallInt	整型变量，范围-32767~32767， 2 字节
Integer	整型变量，范围-20 亿~20 亿， 4 字节
Float	实型变量， 8 位 IEEE 格式
String	可变长字符串变量，最长 32767 个字符。
String*n	定长字符串变量，n 表示字符串长度（最长 32767 个字符）
Logical	TRUE 或 FALSE
Date	日期型
Object	图形对象，如直线或圆
Alias	列变量，用于读表的列值
Pen	线型设置
Brush	填充设置
Font	字体（文本）类型设置
Symbol	符号（点）类型设置

4. 固定长度和变化长度的字符串变量

MapBasic 支持定长和变长两种类型的字符串变量。可变长度的字符串变量能存取不超过 32767 个字符的任何长度的字符串值；固定长度的字符串变量则有一特定的长度极限值，该值用 Dim 语句指定。

声明可变长度字符串变量时，用 String 作为变量类型；声明固定长度字符串变量时，在 String 后面加“ * ”号和允许的字符串最大长度。下面的例子中， full_name 声明为一可变长度的 String 变量，而 employee_id 声明为一长度固定为 18 的字符串变量：

```
Dim full_mame As String,
employee_id As String*18
```

MapBasic 自动地用空格填充固定长度的字符串变量末尾，因此该类变量始终都占有分配的内存空间。若申请了一长度为 5 的固定长度字符串变量，并将“ ABC”赋给该变量，此时变量的实际值是字符“ ABC”，再加后面 2 个空格，这对于编写一个需按格式输出的应用程序非常有用。

5. 数组

MapBasic 数组遵循如下规则：

(1) MapBasic 仅支持一维数组。

(2) MapBasic 中，数组第一个元素对应的下标始终是 1。

(3) MapBasic 应用程序运行在 Windows 3.1 下时，数组最多有 7000个元素； Windows95 下，最多有 32767 个元素。

声明数组变量的方法是在变量名后用圆括号标明数组的大小，这个数组的大小必须用正整数（不允许使用变量）表示。下面的 Dim 语句声明了一个包含 5 个字符串的字符串型数组。

```
Dim string_name(5) As string
```

当数据数组被定义时，各元素的初值是 0：字符串数组各元素的初值是空。

访问数组中的元素采用下面的方法：

```
array_name(element_number)
```

这样，下面的语句赋一个值给 string_name 数组的第 1 个元素：

```
string_name(1)= "mary"
```

要改变数组的大小，使用 ReDim 语句。有些时候不知道会增加多少数据，但又要对它进行处理，这就可以使用 ReDim 语句重新定义数组大小。使用 UBound()函数能确定一个数组的大小。

下面的范例申请一个名为 name_list 的 String 数组变量。其后的程序把数组扩大了 10 个元素。

```
Dim Counter As Integer，name_list(5) As string.
...
counter=UBound(name_list)
ReDim name_list (counter+10)
```

6. 自定义数据类型（数据结构）

在 MapBasic 语言中，使用 Type...End Type 语句定义自定义数据类型。自定义数据类型是一个或多个的变量类型的集合，一旦定义了一数据类型，在后面的程序中就可以通过 Dim 语句声明这种类型的变量。下面的程序定义了一个 employee 数据类型，然后用 Dim 声明 employee 这种类型变量，也就是定义完数据类型后，还须进行声明。

```
Type employee
        name    As String
        title     As String
        id        As Integer
End Type
Dim mamager，  staff(10) As employee
```

自定义数据类型的每一部分都可以看作一个元素。在前面的例子中，数据类型 employee 有 3 个元素：name、title 和 id。为存取该类型变量中的一个元素，使用以下方法：

```
variable_name.element_name
```

下面的过程给变量 manager 的每一元素赋值，

```
manager.name="Teny"
manager.title="Directer"
manayer.id=1234567
```

也可以将该类型声明为一个数组变量，下面给 staff 数组赋值：

```
staff(1).name="Ed"
staff(1).title="Programmer"
```

使用 Type...End Type 语句定义的数据类型，必须事先定义，也就是说 Type...End Type 都出现在程序的最前面。Type...End Type 定义也可以包括其他类型的元素，甚至在此之前用 Type...End Type 定义的数据类型，也可以自定义数据类型的全局变量。

7. 全局变量

用 Dim 定义的变量是局部变量。局部变量仅可以在定义它的过程中使用。 MapBasic 也能定义全局变量，它能在程序的任何过程中使用。

定义一个全局变量，使用 Global 语句。Global 的用法与 Dim 相同，只是用 Global 关键字代替 Dim。下述 Global 语句定义了 2 个全局整型（Integer）变量：

```
Global first_row，last_row As Integer
```

Global 语句必须在任何过程之前定义，所以 Global 总是出现在程序的前面。

下述程序定义了几个全局变量，且这些变量被其他过程所使用。

```
Declare Sub Main
Declare Sub initialize_globals
Global gx，gy As Float
Global Start date As Date
Sub Main
        Dim x，y，    As Float
          Call initialize_globals
End Sub
Sub initialize_globals
        gx=-1
        gy=-1
        start_date=CurDate()
End Sub
```

在编程时，应该尽可能使用局部变量代替全局变量，因为在程序运行的整个过程中，每一个全局变量都占用内存，但是局部变量仅在定义它的过程且该过程在运行时才占用内存。

MapBasic 全局变量也可以使用其他软件包改变它的数据。当应用程序在 Windows 下运行时，其他的应用程序能使用 DDE（Dynamic Data Exchange ）来获得和改变 MapBasic 全局变量的值。

8. 变量的作用范围

过程定义的局部变量可以和全局变量同名。如果局部变量和全局变量名相同，过程不能改变全局变量的值。在过程中，能改变的变量仅是局部变量。

遇到一可使用的变量名时， MapBasic 首先按局部变量对待；如果没有这个局部变量，MapBasic 再按照全局变量处理；如果没有这个全局变量，MapBasic 会尝试为一个打开的表，如果也失败了，MapBasic 就给出错误信息。

3.1.3 表达式和运算符

表达式是一个或者多个变量、常量、函数调用、表存取和操作符构成的集合。

1. 常量

常量的值在程序运行过程中不能改变，也称为硬代码。

数字型常量：不同类型的数字型变量的赋值需要不同类型的常量，如 10，123.45等 。

```
counter=20
```

十六进制数字常量：MapBasic4.0 支持十六进制数字常量，用法与 Visual Basic 相同。其定义是：&Hnumber(number 是十六进制数)。下面的例子将十六进制值 10(等于十进制的 16)赋给变量：

```
Dim i_num As Integer
i_num=&H10
```

数字常量不能有逗号（“，”）（千位的分隔符），下面的表达式是错误的：

```
Counter=1，250，000
```

若数字常量有十进制数的小数点（十进制分隔符），该分隔符必须是句点（.），即使是用户计算机定义了使用其他的十进制分隔符也是如此。

字符串常量：字符串型常量是成对的双引号括起来的部分，如：

```
last_name="Tony"
```

所有字符串型常量不允许超过 256 个字符。

标识符双引号不是字符串型常量的一部分，它们只是标识字符串型常量的开头和结尾，若在字符串型常量中出现必须使用双引号（""）（可以在字符串中插入 2 个双引号），如果需要特殊字符，用 Chr$() 函数。下面的程序说明了如何在字符串中嵌入双引号：

```
Note"The table""world""is already open"
```

逻辑型常量：逻辑型常量只能是 1（true）或者 0(false)。许多 MapBasic 程序都使用 true 和 false 来表示逻辑值。而 true 和 false 的准确定义在 MapBasic 的标准定义文件 mapbasic.def 中，因此要使用像 true 和 false 这样的标准定义，程序必须引用 Include 语句来包含 mapbasic.def。

日期型常量：定义日期型常量，其方法是采用 YYYYMMDD 格式输入 8 位整数。

下例程序指定的日期是 1999 年 12 月 31 日。

```
Dim d_enddate As Date
d_enddate=19991231
```

同样，也能用字符串表示日期型常量：

```
d_enddate="12/31/1995"
```

用字符串定义日期型常量时，年份可以由 4 位或 2 位组成：

```
d_enddate="12/31/95"
```

也可以省略年份，其缺省的年份是当前年。

```
d_enddate="12/31"
```

注意，使用字符串表示日期型常量有时是不可靠的，因为这样所获得的结果取决于用户计算的配置：若用户计算机的日期配置格式是 Month/Day/Year，“06/11/95”表示 6 月 11 口，若格式是 Day/month/year，那么“ 06/11/95 ”表示的是 11 月 6 日。若用户计算机的分隔符是“ - ”，MapInfo 将不能把像“ 12/31 ”这样的字符串表达式转换成日期。

为保证获得唯一的结果，可以使用函数 Number To Date()，它接受 8 位数字的日期格式（数字日期常量，如 19951231），不受用户计算机配置的影响。若必须用字符串作为日期值——如从文本文件中获取日期值——使用 Set Format 语句来控制选择分隔字符串。

Alias 常量：Alias 变量在以后的章节中与表一起详细讨论。

下表给出了各种常量的例子：

类 型	赋值范例
Integer	i=1234567
SmallInt	m=90