

应用开发数据库

汉字 DATA FLEX

宫浦 心 旭宏 译

刘胡 增 露炎 校

中国科学院计算所中计公司

中国科学院计算所八室

中国科学院信通电脑公司

译者前言

DataFlex是美国Data Access公司和Digital Research公司联合研制的新型应用开发数据库。自1981年推出以来，进行了不断的版本更新和升级。DataFlex既有单用户版，也有多用户版，可以运行在多种微机上。

和目前广泛流行的dBASE III相比，DataFlex具有许多不可比拟的优点。例如：引进“像”这一新的概念，可以随意定义输入输出的格式（如表格等），使对输入输出要求很高的用户感到得心应手；设立许多宏命令和自动功能，使编程变得简单；命令语言丰富完整，并且可由用户对它进行扩充；提供了便利的多数据库相关操作，便于用户开发复杂的应用系统；另外还可以和外界进行数据文件交换，等等。DataFlex适用于各种应用系统的开发，相信它在国内一定会得到广泛的使用。

译者对IBM PC上的单用户2.1/b版DataFlex进行了完全汉化，所有提示都改为汉字提示，任何ASCII串中都可包含汉字，文件名、字段名以及变量名也可以使用汉字。另外，汉化DataFlex要在联想汉卡的支持下运行。这是对本书的一点补充。

本书不仅仅是一本手册，同时还介绍了DataFlex所引入的新概念和新方法以及有关重要的数据库知识，很值得一读。初学者要仔细阅读本书的前三章，待动手编制较复杂的程序时可进入后面的章节。

本书的第二章、第三章、第五章和第六章由宫旭同志翻译，浦心宏同志翻译了第一章和第四章。刘露、胡增炎校阅了全稿并提出许多宝贵意见，在此深表感谢。另外，还感谢刘子龙同志帮助整理了部分稿件。

由于译者水平所限，书中难免有错误和不当之处，欢迎读者指正。

译者

一九八七年十一月

译者前言

DataFlex是美国Data Access公司和Digital Research公司联合研制的新型应用开发数据库。自1981年推出以来,进行了不断的版本更新和升级。DataFlex既有单用户版,也有多用户版,可以运行在多种微机上。

和目前广泛流行的dBASEⅢ相比,DataFlex具有许多不可比拟的优点。例如:引进“像”这一新的概念,可以随意定义输入输出的格式(如表格等),使对输入输出要求很高的用户感到得心应手;设立许多宏命令和自动功能,使编程变得简单;命令语言丰富完整,并且可由用户对它进行扩充;提供了便利的多数据库相关操作,便于用户开发复杂的应用系统;另外还可以和外界进行数据文件交换,等等。DataFlex适用于各种应用系统的开发,相信它在国内一定会得到广泛的使用。

译者对IBM PC上的单用户2.1/b版DataFlex进行了完全汉化,所有提示都改为汉字提示,任何ASCII串中都可包含汉字,文件名、字段名以及变量名也可以使用汉字。另外,汉化DataFlex要在联想汉卡的支持下运行。这是对本书的一点补充。

本书不仅仅是一本手册,同时还介绍了DataFlex所引入的新概念和新方法以及有关重要的数据库知识,很值得一读。初学者要仔细阅读本书的前三章,待动手编制较复杂的程序时可进入后面的章节。

本书的第二章、第三章、第五章和第六章由宫旭同志翻译,浦心宏同志翻译了第一章和第四章。刘露、胡增炎校阅了全稿并提出许多宝贵意见,在此深表感谢。另外,还感谢刘子龙同志帮助整理了部分稿件。

由于译者水平所限,书中难免有错误和不当之处,欢迎读者指正。

译者

一九八七年十一月

2. 3. 3	应用程序的索引	(56)
2. 3. 4	AUTODEF程序	(57)
2. 4	应用程序的编制	(60)
2. 4. 1	源文件的建立	(60)
2. 4. 2	应用程序的编译	(61)
2. 4. 3	应用程序的运行	(62)
2. 5	DataFlex 实用程序	(62)
2. 5. 1	编辑程序	(62)
2. 5. 2	重新索引	(64)
2. 5. 3	转换程序	(68)
2. 5. 4	如何把 dBASE II 文件转换成 DataFlex	(68)
2. 5. 5	创立和维护菜单	(71)

第三章 编程准备

3. 1	程序的基本元素	(75)
3. 1. 1	串常量	(75)
3. 1. 2	串变量	(76)
3. 1. 3	数字常量	(76)
3. 1. 4	数字变量	(76)
3. 1. 5	表达式	(77)
3. 1. 6	日期变量	(77)
3. 1. 7	日期常量	(78)
3. 1. 8	文件名	(78)
3. 1. 9	指示变量	(78)
3. 2	变量定义	(79)
3. 2. 1	STRING命令	(79)
3. 2. 2	NUMBER命令	(80)
3. 2. 3	DATE命令	(80)
3. 2. 4	INTEGER命令	(80)
3. 2. 5	INDICATOR命令	(81)
3. 3	参量的处理	(81)
3. 3. 1	MOVE 命令	(81)
3. 3. 2	INCREMENT 命令	(82)
3. 3. 3	CALCULATE命令	(82)
3. 4	数据输入	(82)
3. 4. 1	ENTER 宏命令	(82)
3. 4. 2	ENTER 程序结构	(83)
3. 4. 3	ENTER像部分	(84)
3. 4. 4	ENTER命令行	(84)
3. 4. 5	输入部分	(84)

3. 4. 6	ENTRY命令格式选项	(85)
3. 4. 7	输入和AUTOPAGE	(87)
3. 4. 8	用户定义过程	(88)
3. 4. 9	ENTDISPLAY命令	(89)
3. 4. 10	输入后处理	(90)
3. 4. 11	ENTER宏的执行顺序	(90)
3. 4. 12	指示变量ENT\$PERMISSIVE	(91)
3. 5	报表输出	(91)
3. 5. 1	REPORT宏命令	(91)
3. 5. 2	REPORT宏的各部分	(92)
3. 5. 3	子合计与合计	(93)
3. 5. 4	选择	(97)
3. 5. 5	页控制	(99)
3. 5. 6	有关REPORT宏的注意事项	(103)
3. 6	文件定义	(104)
3. 6. 1	设计数据文件间的关系	(104)
3. 6. 2	如何建立关系字段	(106)
3. 6. 3	列出各个字段	(107)
3. 6. 4	复盖型字段	(108)
3. 6. 5	确定关键字段	(109)
3. 6. 6	使用FILEDEF	(110)
3. 6. 7	创立/编辑字段	(112)
3. 6. 8	创立/编辑索引	(114)
3. 6. 9	印出文件定义	(116)
3. 6. 10	置文件参数/名字	(117)
3. 6. 11	删除数据文件	(117)
3. 6. 12	置文件休眠	(117)
3. 6. 13	根据屏幕像创立文件定义	(117)
3. 6. 14	根据DEF文件创立文件定义	(117)
3. 6. 15	保存定义/退出FILEDEF	(117)
3. 6. 16	中止FILEDEF, 不更新文件定义	(118)
3. 6. 17	如何使用FILEDEF更改现存文件定义	(118)
3. 7	应用举例	(119)

第四章 DataFlex程序设计

4. 1	与“像”有关的输入及输出	(134)
4. 1. 1	PAGE命令	(134)
4. 1. 2	ACCEPT命令	(134)
4. 1. 3	DISPLAY命令	(135)
4. 1. 4	AUTOPAGE命令	(135)

4. 1. 5	CLEARFORM 和 BLANKFORM 命令	(136)
4. 1. 6	OUTFILE命令	(137)
4. 1. 7	OUTCLOSE命令	(137)
4. 1. 8	OUTPUT 命令	(138)
4. 1. 9	NAME命令	(138)
4. 1. 10	FORMAT 命令	(138)
4. 1. 11	PRINT 命令	(140)
4. 1. 12	ENTERGROUP	(140)
4. 1. 13	CURRENT—WINDOW和CURRENT—IMAGE	(143)
4. 2	串操作	(143)
4. 2. 1	串操作的注意事项	(143)
4. 2. 2	LEFT命令	(144)
4. 2. 3	RIGHT 命令	(144)
4. 2. 4	MID命令	(145)
4. 2. 5	ASCII 命令	(145)
4. 2. 6	CHARACTER命令	(145)
4. 2. 7	POS 命令	(146)
4. 2. 8	LENGTH 命令	(146)
4. 2. 9	PAD命令	(147)
4. 2. 10	TRIM 命令	(147)
4. 2. 11	INSERT 命令	(147)
4. 2. 12	REPLACE 命令	(148)
4. 2. 13	APPEND 命令	(148)
4. 2. 14	UPPERCASE命令	(149)
4. 2. 15	CMDLINE命令	(149)
4. 3	条件执行	(149)
4. 3. 1	指示变量	(150)
4. 3. 2	设置指示变量的状态	(151)
4. 3. 3	系统预先定义的指示变量	(153)
4. 3. 4	IF测试	(154)
4. 3. 5	使用NOT	(156)
4. 3. 6	用于文件的 STATUS 指示变量	(156)
4. 4	控制执行顺序	(156)
4. 4. 1	GOTO命令	(157)
4. 4. 2	ON GOTO 命令	(157)
4. 4. 3	GOSUB 命令	(157)
4. 4. 4	ON GOSUB命令	(157)
4. 4. 5	RETURN命令	(158)
4. 4. 6	CHAIN 命令	(158)

4. 4. 7	DEBUG命令	(159)
4. 4. 8	ERROR命令	(159)
4. 4. 9	ON ERROR命令	(160)
4. 4. 10	CLEARWARNING 命令	(161)
4. 4. 11	ABORT命令	(161)
4. 5	程序块和循环	(161)
4. 5. 1	BEGIN—END命令	(162)
4. 5. 2	REPEAT—UNTIL 命令	(163)
4. 5. 3	REPEAT—LOOP命令	(163)
4. 5. 4	WHILE—END 命令	(164)
4. 5. 5	FOR—FROM—TO—LOOP 命令	(164)
4. 6	控制台的直接控制	(165)
4. 6. 1	SHOW命令	(165)
4. 6. 2	SHOWLN 命令	(165)
4. 6. 3	INPUT 命令	(166)
4. 6. 4	GOTOXY 命令	(166)
4. 6. 5	CLEARXY命令	(167)
4. 6. 6	CLEARSCREEN命令	(167)
4. 6. 7	KEYCHECK命令	(167)
4. 6. 8	INKEY和INKEY\$ 命令	(168)
4. 6. 9	SCREENMODE 命令	(168)
4. 7	对记录的操作	(168)
4. 7. 1	数据库命令和FILEDEF (文件定义)	(168)
4. 7. 2	了解记录缓冲区	(169)
4. 7. 3	文件操作	(169)
4. 7. 4	OPEN命令	(170)
4. 7. 5	FIND 命令	(171)
4. 7. 6	SAVE命令	(172)
4. 7. 7	SAVERECORD 命令	(173)
4. 7. 8	CLEAR命令	(173)
4. 7. 9	DELETE 命令	(174)
4. 7. 10	RELATE命令	(174)
4. 7. 11	ATTACH命令	(174)
4. 7. 12	ZEROFILE命令	(175)
4. 7. 13	系统文件	(175)
4. 8	对文件的操作	(175)
4. 8. 1	ERASEFILE命令	(176)
4. 8. 2	RENAMEFILE 命令	(176)
4. 8. 3	COPYFILE 命令	(176)

4. 8. 4	DIRECTORY 命令	(176)
4. 8. 5	FILELIST 命令	(176)
4. 8. 6	FILELIST NEXT命令	(177)
4. 8. 7	FILELIST PATHNAME 命令	(177)
4. 8. 8	FILE—SIZE 命令	(177)
4. 8. 9	INDEX—DEF 命令	(178)
4. 8. 10	MAKE—FILE 命令	(178)
4. 8. 11	MEMORY 命令	(179)
4. 8. 12	SYSTEM 命令	(179)
4. 8. 13	RUNPROGRAM 命令	(180)
4. 8. 14	REGISTRATION命令	(180)
4. 8. 15	SYSDATE 命令	(180)
4. 8. 16	FIELD—DEF 命令	(181)
4. 9	直接输入输出	(181)
4. 9. 1	DIRECT—INPUT 命令	(181)
4. 9. 2	DIRECT—OUTPUT和 OUTFILE 命令	(182)
4. 9. 3	CLOSE—INPUT和CLOSE—OUTPUT 命令	(182)
4. 9. 4	READ命令	(182)
4. 9. 5	READLN 命令	(183)
4. 9. 6	预先定义的指示变量 SEQEOF和SEQEOL	(184)
4. 9. 7	READ—BLOCK命令	(185)
4. 9. 8	WRITE 命令	(185)
4. 9. 9	WRITELN 命令	(186)

第五章 高级程序设计

5. 1	使用功能键	(187)
5. 1. 1	使用 Flex-键的指示变量	(188)
5. 1. 2	使用键控程序	(188)
5. 1. 3	ENTAGAIN 命令	(190)
5. 1. 4	ENTERMODE 命令	(190)
5. 1. 5	BACKFIELD 命令	(191)
5. 1. 6	HELP命令	(191)
5. 2	多用户功能	(193)
5. 2. 1	ENTER宏的多用户操作	(193)
5. 2. 2	LOCK命令	(194)
5. 2. 3	UNLOCK 命令	(194)
5. 2. 4	REREAD 命令	(194)
5. 2. 5	IFCHANGE命令	(194)
5. 2. 6	SETCHANGE命令	(195)
5. 2. 7	DESPOOL 命令	(195)

5.3	字段索引和窗口索引.....	(195)
5.4	DataFlex 的修改和扩充.....	(197)
5.4.1	替换.....	(197)
5.4.2	命令定义.....	(198)
5.4.3	包含文件.....	(199)
5.4.4	编译时变量.....	(199)
5.4.5	条件选择.....	(199)
5.4.6	符号的内部类型.....	(201)

第六章 附录

6.1	DataFlex 的保留字.....	(201)
6.2	文件、参量和命令.....	(205)
6.2.1	文件名及扩展名类型.....	(205)
6.2.2	预定义的参量.....	(206)
6.2.3	窗口格式和输入选项.....	(207)
6.2.4	命令语法列表.....	(208)
6.3	系统描述.....	(212)
6.4	词汇.....	(214)
6.5	ASCII字符表.....	(218)
6.6	编译错误信息.....	(219)
6.7	运行错误状态和诊断.....	(223)
6.7.1	错误状态类型.....	(223)
6.7.2	运行状态分类.....	(223)
6.7.3	OP: 操作员错.....	(225)
6.7.4	MH: 介质和硬件错.....	(226)
6.7.5	L/D: 磁盘限制.....	(228)
6.7.6	L/M: 内存限制.....	(228)
6.7.7	L/C: 程序限制.....	(230)
6.7.8	CE: 程序错误.....	(230)
6.7.9	SU: 设置错误.....	(232)
6.7.10	多用户状态.....	(233)
6.7.11	用户错误状态信息.....	(234)

第一章 操作与指南

1.1 本手册的组织

1.1.1 介绍的顺序

本手册将按学习和使用DataFlex的先后次序来组织教材,以使读者能尽快地把学到的经验与知识转变成实践经验。只需几个小时,你就可以初步具备建立数据库结构并用程序(Configuration)加以控制的能力。甚至就在你开始做练习时,就能编制简单明了的应用程序。尽管简单,但功能却很强,它可以高效地完成任意数量的一般数据处理任务,如邮政列表等。这里所说的程序(Configuration)是指用DataFlex书写的数据处理过程。

但你非常希望得到DataFlex的根本原因,是因为它的强大功能和灵活性,而不只是它的单纯表格处理。当你继续阅读本手册,并试用每部分所描述的功能时,会遇到越来越多的问题并且复杂性也随之迅速提高,这样由浅入深,直至你掌握所有数据管理所需要的手段为止。一般说来就DataFlex的简单功能已经是非常强大了,所以,一般情况下,就不必探究后面章节中所描述的DataFlex的更高级的功能了。除非你想要熟悉了解DataFlex的更高级的功能特色。

1.1.2 本手册的符号约定

本手册中所使用的符号都有一定的约定,为了便于理解手册中给出的格式和例子,应该首先了解这些约定。DataFlex程序由屏幕像(总是在程序开头位置)和随后的命令行组成,由独占一行的组合字符“/•”将这两部分隔开。DataFlex命令就象是一个动词(命令)和(通常)一个或多个宾语(被称为参量)组成的句子,有时后面还跟着一个或多个任选项,有时还跟着一个注释。命令前面可以放置一个称为指示变量的条件项。最后,某些命令行以“标号”开始,由一个冒号“:”终止,以区别于命令本身。其用途是为将来在程序中引用此段程序(如RETURN,GOTO命令)“标记”一个入口点。下面是一个命令行的典型格式:

[指示变量]命令 参量 {任选项} //注释

上述命令行中的方括号[]是用以识别指示变量的,而大括号{}在大多数命令中并不需要。ENTRY和FORMAT命令的格式任选项除外,那里的大括号{}是必需的。大括号在格式说明中,用以表示一个任选的命令元素。再者,除专门说明以外,它们是不书写在命令中的。放在注释前的双斜线符号“//”就象指示变量中的方括号一样,是DataFlex句法的一部分,并告诉DataFlex不用去编译跟在双斜线符号后面的信息,这些信息只保留在源程序中,以便别人阅读。如在一行的开头出现双斜线符号,则使整个命令都成为注释。下面是一个典型的命令行:

[FOUND] ENTRY mail.name {CAPSLOCK} //第一次尝试

该行的开始是由方括号标识的一个DataFlex指示变量FOUND,后面跟随命令ENTRY,然后是参量,该参量是在程序中已经命名的屏幕像的窗口(名为mail.name),再跟一个格式化任选项CAPSLOCK(这里需要大括号)。本命令行的意思是:“如果在数据库中成功地查找到所期望的记录,则把光标放入下一个屏幕像窗口,并把ENTRY将在窗口中键入的字符接收到数据库元MAIL.NAME中,而且就好像CAPSLOCK键处于接通状态(输入的小写字符

全部转换成大写字符”。注释（第一次尝试）不由DataFlex解释，因而对执行程序毫无意义，只便于阅读程序。

上述例子中对大写和小写字母的使用说明，适用于本手册中的所有例子。FOUND，ENTRY及CAPSLOCK都用大写表示，而mail.name却以小写字母出现。这是为了着重表明FOUND，ENTRY及CAPSLOCK是DataFlex的固有部分，并分别匹配DataFlex的一个指示变量，一个命令和一个任选项。凡是用这些元素书写的任何程序都要使用这些字符串。另一方面，mail.name是所讨论的例子或程序所特有的。不同的人书写的任何程序都可以根据其习惯使用其它特定的串（当然，也可以使用与上述相同的串）。你还会发现，在本手册中的所有格式或例子都是照此方式使用大写和小写字母的，但完整的样本程序除外，在这种情况下，按通常书写程序的习惯，全部使用大写字母，但同时需要大、小写的屏幕提示正文资料和信息除外。

在本系统及手册中，单引号（'）或双引号（"）所包含的所有DataFlex的资料都是可以在程序中传递、显示和输出但不能用来执行的文字资料。

除以上所述的约定以外，DataFlex本身还有许多句法约定，它们在本手册中有特殊的表示方法。对此，将在本手册的适当地方分别予以介绍。

1. 2 操作指南

本节介绍用DataFlex开发应用系统时所涉及各个步骤，以及在系统建立和运行时要用到的各种文件和程序。

DataFlex为开发应用系统提供各种各样的功能，使用者为实现某个特定的目标，往往有多种方法或途径。这样，你就可以从中选择一种最佳的实现方法。所以，在实践时，一定要避免认为你设想的第一种方法是唯一的或最好的。就是说应该多方考虑，看是否可能有别的更容易、更有效的途径。

本节分成两部分：运行时系统和应用开发。所谓运行时系统是指为执行（运行）应用程序所必需的DataFlex部分，运行程序的人称作“操作员”（以区别开发应用程序的程序员）。应用开发部分描述的是支持应用系统开发的部分。尽管程序员和操作员在某些情况下可能是同一个人，但编制程序和操作是二种不同的活动，所以弄清这些功能之间的差别是有益处的。

1.2.1 安装

DataFlex的安装主要包括将必要的文件从DataFlex主盘复制到工作盘或硬盘上去，然后使用DataFlex的SETSCREEN程序，根据顺序号和终端类型对系统初始化。SETSCREEN程序读文件TERMLIST.CFG，并将由用户输入的终端信息写到运行时系统和程序系统要用的FILELIST.CFG文件中。有关安装的更详细介绍请参见安装说明及“安装DataFlex”部分。

1.2.2 运行时系统

DataFlex运行时系统的基本操作核心是程序文件“FLEX.COM”（在CP/M—86系统上是FLEX.COM）。FLEX.COM及其复盖文件“RUN.*”是在整个DataFlex程序运行过程中都起作用的文件。DataFlex程序具有令计算机完成特殊任务的智能。对运行时系统来说，若无编译好的程序文件（.FLX），则它将一事无成。

在程序控制下, DataFlex 运行时系统可以访问并对作一个或多个数据库进行操作, 和操作员进行交互对话、驱动打印机、读写标准 ASCII 文件并且具有一般微机所具有的各种功能, 这些动作由开发者建立和编译的程序(带有“.FLX”扩展名的文件)来控制。当 FLEX 由操作系统运行时, 必须给出要运行的程序名。

在初始化时, DataFlex 运行系统首先要读一个 FILELIST.CFG 文件, 该文件包含有关的可用数据库文件的信息, 以及允许 DataFlex 生成特殊终端的控制码。

FILELIST.CFG 中出现的每个数据库文件名实际上都属于磁盘上相关文件的“家族”, 它们都有共同的根名, 但有不同的扩展名。数据文件(其扩展名为“.DAT”)包括按输入顺序排列的原始数据, 及数据文件结构的基本信息。关键码文件(“.Kx”, 扩展名中的 x 是索引编号)包含有索引数据, 它提供根据定义数据库过程中建立的索引存取数据的方法。标签信息文件(扩展名为“.TAG”)包含组成数据库的字段名称。

因此, 运行 DataFlex 所需要的最小文件组合是: FLEX.COM, RUN.OVF, FILELIST.CFG, 至少有一个由程序员预先建立好的.FLX(程序)文件, 以及在程序(所用的)中说明的数据库的所有数据库文件(.DAT, .Kx 及 .TAG)。为了使用查询(QUERY)功能, 则 QUERY.* 程序文件也必须存在。

当 DataFlex 初始化时用户所见到的 DataFlex 菜单系统本身也是一个 DataFlex 程序, 菜单程序的名称是 MENU.FLX, 菜单程序把菜单保存在一个名为 MENU.DAT 的标准 DataFlex 数据库文件中, MENU.DAT 作为 49 号文件安装在 FILELIST.CFG 提供的活动数据库表中。因此, 为使用菜单功能, 系统中必须有 MENU.FLX 和 MENU.DAT。

在运行时, 还可能偶然地要求提供其他程序。REINDEX 读取一个数据文件并重建关键码文件(.Kx 文件), 检查(如果需要并重建)数据库文件中删除的记录表, 并删除或提出已遭破坏(典型的情况, 如掉电, 介质或系统破坏)的记录。REINDEX 也用于维护 BATCH 索引以及建立“附加(ad hoc)”索引。该程序若不需要, 则一直维持脱机状态。

1.2.3 应用开发

用 DataFlex 开发特定的应用系统, 其过程包括: 建立程序(带有.FLX 扩展名的文件)和建立按所需要的方式存贮和操作数据的数据库文件, 这些文件通过使用由 DataFlex 所提供的一组实用程序来建立。“.FLX”文件通过对“源”文件运行 DataFlex 编译程序来建立。建立源文件可有多种不同的手段, 可任选一种。如 DataFlex 所提供的 EDITOR.COM 程序, 或能生成普通 ASCII 正文文件的任何文本编辑或字处理器。源文件包含 DataFlex 的各种命令, 这些命令可以使系统完成所希望的各种处理过程。

一般初学者往往都用 AUTODEF 实用程序来产生自己的第一个程序。AUTODEF 还为新数据库创立定义, 及其数据所必需的数据和关键码文件。使用该程序的第一步是“描绘”一个输入和查询的屏幕像, 该屏幕像将在新数据库中使用(参看“像的格式”一节)。DataFlex 的 EDITOR 或你所熟悉的正文编辑器将承担这一工作。AUTODEF 读该文件并产生一个文件定义(.FD)文件, 以及一个程序源代码文件(带.FRM 扩展名), 以准备对它进行编译。由 AUTODEF 产生的程序可以完成它所建立的数据库文件的所有文件维护。

EDITOR 程序的功能是建立、显示和修改 DataFlex 源程序和像文件。源文件可以通过 EDITOR 建立, 或来之于类似 AUTODEF, QUERY 或 READ 等自动定义的某个实用程

序。由EDITOR产生的源文件是标准ASCII文件，因此，根据使用者的爱好与习惯，可以选用大多数字处理器和编辑器而不用DataFlex的EDITOR。

DataFlex的QUERY程序为用户访问磁盘驱动器上现有数据库中所存储的信息提供了方便。QUERY程序可以用作附加报表生成器，且稍加训练即会使用。QUERY完全是提示驱动方式，因此，对操作员来说非常实用，但一旦完成一次成功的查询，它将允许把查询标准作为报表程序的源代码存储在一个文件中。然后对源文件进行编译，生成一个永久的报表生成程序。QUERY要求提供所有标准的运行时数据库。

DataFlex的READ程序产生一个用于引进标准ASCII数据文件的源程序。READ主要为实现从其他系统到DataFlex的数据转换而设计的。

REINDEX程序(REINDEX.COM或REINDEX.CMD)检测数据文件中的非ASCII字符，并识别出被破坏的记录以便将其去除或纠正，允许对任何数据库的附加文件索引进行操作，以及为由于掉电、不适当的程序动作等原因造成索引被破坏后提供改变索引定义和重建现存索引文件的功能。

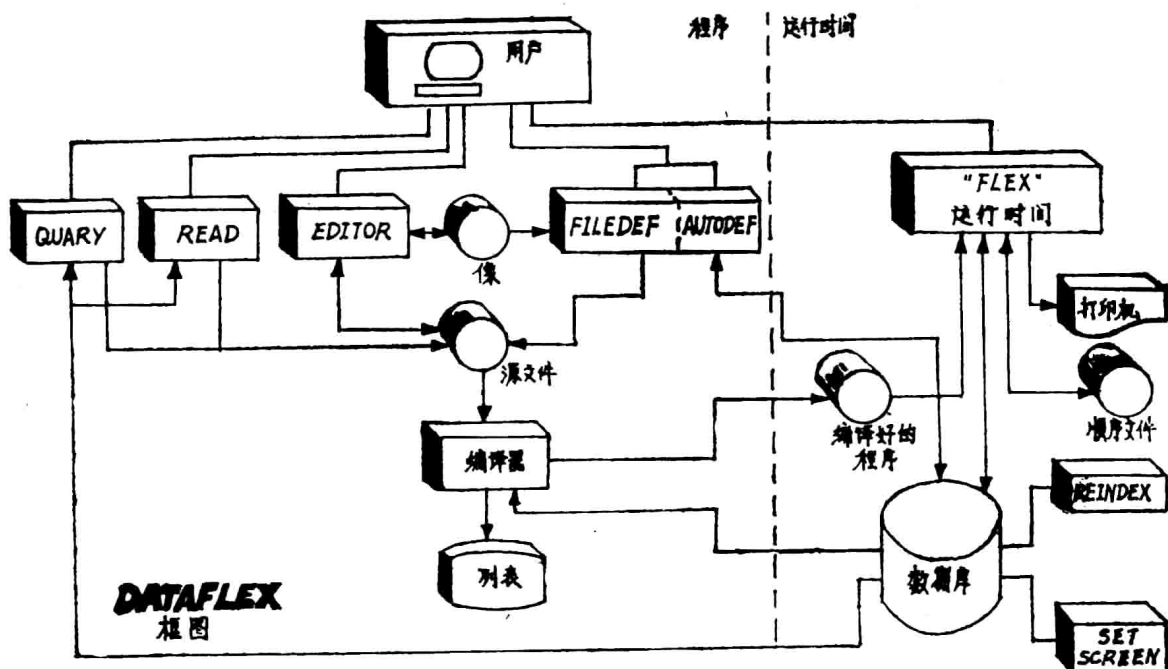
从源文件中产生.FLX文件的编译器(COMP.COM或COMP.CMD，以及8位机系统的COMP.OVF)，为完成它的工作需要三个文件存在于登录盘驱动器上：源程序文件、程序中用(说明)的每个数据库的文件定义(.FD扩展名)文件、以及DataFlex命令宏文件(FLEX.CFL)，“.FD”文件和FLEX.CFL必须都驻留在登录(缺省)盘驱动器上。一次成功的编译将产生一个具有.FLX扩展名，同时又和源文件具有相同根名的可运行的DataFlex程序。当编译器运行时屏幕上将显示原始源命令和编译器所赋与的行号，以及编译时可能出现的错误信息。这些信息的输出可以被定向到三种设备的任何一种设备上：控制台，打印机或者一个可打印的(.PRN)文件。

不使用数据库的DataFlex程序，在编译时不需要任何.FD文件，但一个典型的DataFlex程序将存取一个或多个数据库。“.FD”文件仅在编译时使用，运行时是不需要的。另一方面，其它运行时数据库文件(.DAT, .Kx)在编译时不需要。这种不同阶段对不同文件的需要，在缺少空间时会显出其优点。

FILEDEF实用程序就象AUTODEF一样，可用来建立数据库文件和程序源文件。但FILEDEF比AUTODEF更灵活。FILEDEF可用来建立多文件关系。使用FILEDEF需要更多的有关DataFlex的知识，但能对数据库给出更多的控制。FILEDEF程序也能读写标准ASCII的数据定义(.DEF)文件。定义文件对于引用、建立数据库文档以及传送应用系统到其它环境中去都是有用的。

当用户想建立自己的新命令，并把它们加到DataFlex语言中去时，可以使用PACK程序。PACK程序读文件FMAC并创立编译器所使用的文件FLEX.CFL。新命令宏可以用文件编辑器或字处理器加到FMAC中去，然后对新版本FMAC运行PACK以建立起一个可用的新版本FLEX.CFL(重写老版本)。

运行SETSCREEN并将操作终端的代码“告诉”DataFlex之后，DataFlex就有三种基本的使用级别可供选用。只要循序渐近的使用，就能最容易、最完全地掌握DataFlex的使用经验。本手册的内容就是按这个次序提供的，这三个级别分别介绍如下：



(1) 查询和报表

用户DataFlex 系统已经提供了一组由示例数据实现的工作程序,全部存放在盘上,并已编译完毕可以运行。这些示例只用于辅助学习,所以在本手册中未加详述。其中多数可以从“DataFlex 表演文件”子菜单中找到,进入子菜单可以通过主菜单选择完成。实际上,通过该菜单可以运行任何程序,并(或者)可通过选择“查询数据文件”选择项来查询它们的数据库。这些示例,用户可以自由试用,输入附加的测试数据并生成报表送到屏幕或打印机上显示出来。本手册的操作员指南将提供该项功能所需要的全部指导。最后,甚至可以修改这些程序,使其按用户的需要按不同方式运行(某些程序实际上可能非常接近于用户所需要的功能)。每个程序都在C: 盘上提供有源“代码”(这是因为要改变程序就必须要对它进行修改并重新编译)。

(2) 提示化的数据库创立和程序

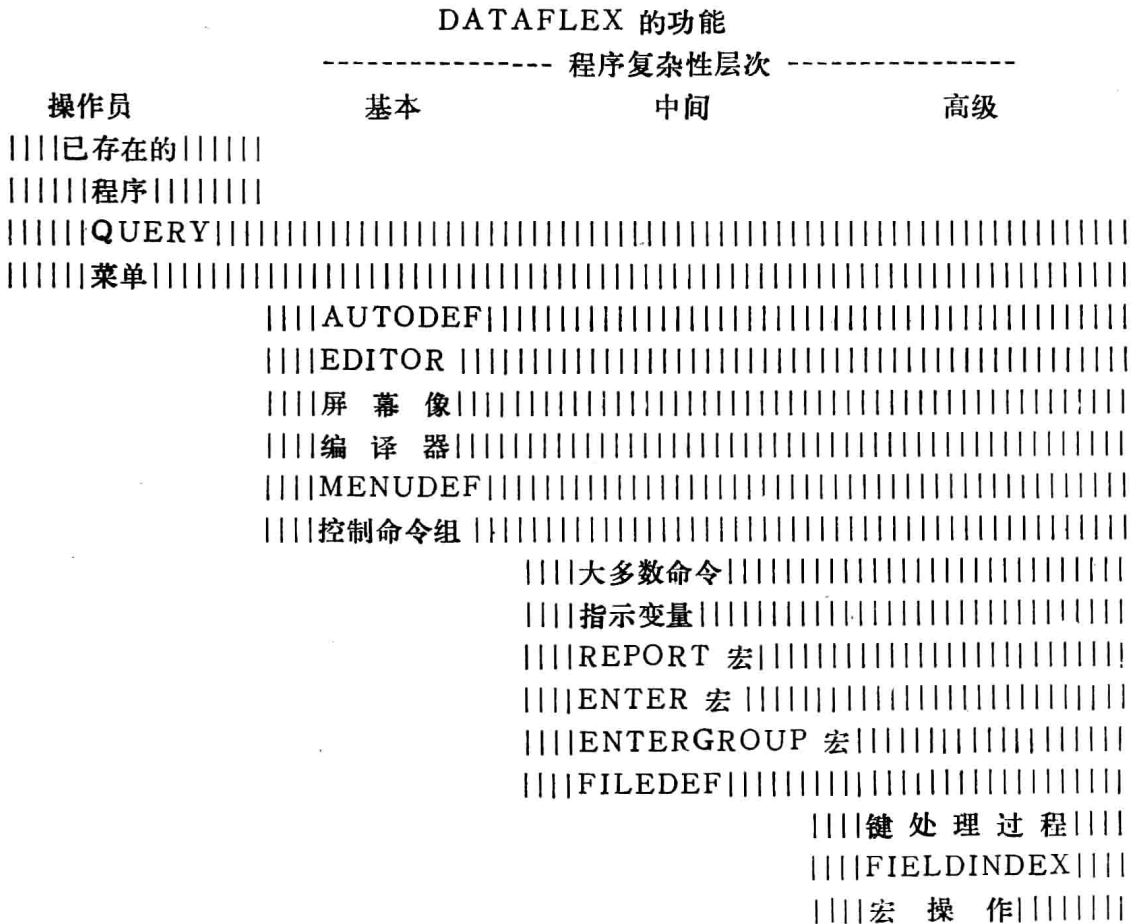
DataFlex 对于建立和维护数据库,以及建立向数据库输入数据和从数据库中产生报表的程序提供了许多“自动的”方法。而“捷径”方法都是提示化的,其结构比书写DataFlex 程序的“程序设计”方法更严密,对此,切不可低估。实际上,它们是高度灵活而又非常强大的。大体上讲,四个DataFlex用户中起码有两个用户只需使用EDITOR(用于书写程序),MENUDEF(用于建立和编辑菜单),AUTODEF(用于建立数据库和输入程序)和QUARY(用于产生报表及完成交互方式的查询)就足够了,而很少需要使用其它DataFlex 功能。上述四个有代表性的用户中的其他两个用户,也仅有一个用户要求不能由上述功能及ENTER宏和REPORT 宏所满足的功能。应当充分考察上述功能。只有在有独特的需求或某种迫切的需要,以至于应用当前市场上所能提供的众多灵活的应用开发工具都不能满足时,方可进入下一级。

(3) 程序设计和语言的扩展

按逐行方式进行应用程序设计时, DataFlex为创立应用系统提供了连创立者都想像不

到的功能。功能键命令为用户提供了更多的基本功能，如顺序输入/输出，控制台输入/输出以及多用户功能可允许用户构造任何实际所希望的处理结构。虽然，整个程序都可以由这些命令构成，但这些命令更有可能和那些由诸如REPORT及ENTER宏等预编程功能所建立的结构混合使用。然而，清一色地仅用这些功能来书写程序可能是建立程序的低效手段，因此，必要时利用这些命令来改进那些标准的结构功能将是一种大受鼓励的方法。在本手册的高级部分，你甚至可以看到修改DataFlex命令本身以及建立为供个人使用的全新命令的方法。

下图对DataFlex 主要应用功能中每项功能的程序设计复杂性范围作了更直观的图形描述。图表中没有过多的文字解释，也没有说明为什么这样的理由，例如，QUERY能够满足应用系统的某个特殊功能的要求，又为什么不在其他更高级的应用中利用QUERY的基本功能来建立报表呢？进而言之，建立高度复杂应用系统的共同方法是，用“基本”或“中间”功能（如：QUERY或AUTODEF）建立应用系统的“核心”，然后再对核心进行编辑，加入应用所要求的更高级的功能特征。



1. 3 操作员指南

这一部分的目的是提供一个清晰的、非技术性的有关数据输入系统部分的使用摘要。本节的第一部分将定义使用DataFlex进行数据输入和文件维护的过程，其次将讨论在任何给

定的应用系统中可以使用的特性和功能。

DataFlex使用一套称为“Flex-键”的命令键去执行应用软件系统中的许多经常使用的功能。Flex键的一个例子是FIND，用于在数据库中定位一个记录，还有SAVE，用于在数据库中存贮一个新记录，等等。而且由于DataFlex能在非常广泛的不同系统和各种特殊的终端中得到应用，所以我们在讨论的过程中，将用Flex键的名称而不采用特定终端上所建立的特殊键来引用程序功能，这是因为命令键的位置随系统的不同而不同。

SETSCREEN实用程序用于将DataFlex的 Flex 键命令赋给键盘上的一组控制（或功能）键，该程序操作的一部分可打印出当前终端上所赋与的命令和键的列表。如果还没有那张列表的付本，这时请运行SETSCREEN程序，打印出本节要使用的键列表。

1.3.1 功能综述

包含有数据输入的一个DataFlex应用程序能够从键盘上接受数据，检查数据类型的正确性，然后将它存贮到计算机的软或硬盘上，一旦数据被存贮，DataFlex就能够找到（FIND）数据并将它显示在CRT上，且允许操作员去修改或删除它。

DataFlex的所有功能，在每一个应用程序中都是可用或可不用的。例如，程序中的一个选择项对操作员封锁屏幕上的一个特殊的数据窗口（使之不能输入数据），或者在处理某些数据输入的情况下由程序自动进行计算。这里我们的着眼点尽管是要讨论DataFlex的所有能力，但在你的计算机上运行的每一个应用程序都不一定含有所有这些功能。

1.3.2 术语定义

控制键：当“控制”键被按住时产生的键盘输入，如^A或CTRL A 这样的各种表示方法表示“A”也被按下时控制键一直被按着，可以用SETSCREEN来把Flex键功能赋与控制键。

数据输入：将新的信息通过终端键盘放入计算机的过程。

文件维护：已存贮在计算机中信息的检查，修改或删除。

填充字符：显示在“窗口”（见下）中的一个字符，定义数据字段的长度并使之在一个特定位置上输入和显示。通常都使用下划线字符（-----）。

功能键：在终端键盘上的正常打字排列之外的特殊键，产生一个能被DataFlex 捕获的代码信号以执行某个DataFlex的Flex键功能（如，FIND、DELETE、SAVE等等）。终端功能键产生的字符不在屏幕上显示。

索引：文件内连接到记录的关键码的结构列表。索引由DataFlex 自动维护，这样当数据库中的数据随机地输入和存贮时都能够按照一定的顺序存取其记录。

关键码：关键码是记录中的一个数据元，根据这个数据元就可以访问或查找数据库中的记录。例如，“帐号”以及（或者）帐目名称就能够定义为一条可接受的帐目记录的关键码；“条目标识符”以及（或者）“说明”能够作为库存记录的关键码。

关键码字段：同索引。

窗口：定义为接收或显示数据的一个屏幕显示区。窗口可被定义为只接收一种类型的数据……，例如：只接受数值型。

1.3.3 如何开始

可以选择系统菜单中适当的选项来开始数据输入和文件维护。对于所希望执行的数据输

入程序，需要做的全部工作是从菜单中选择选项号。

DataFlex屏幕是由操作员从键盘上产生的Flex-键命令来操作的。对于特定系统，这些命令可被赋给一定的键（例如：TAB、LINE FEED、DELETE、RUB-OUT等等）、功能键（F₁、F₂、等等）以及（或）控制序列（CTRL A、CTRL B等），这个赋予工作是用称为SETSCREEN的DataFlex实用程序来完成的。一旦命令被赋给了终端上的各种各样的键，则不论要处理的文件或者屏幕格式，所有的DataFlex屏幕都具有完全相同的操作。

由于整个系统具有过程连续性，DataFlex的这个特点可提高操作员的效率并减少出错机会。命令的执行就成为第二性的了，那么当一个新的或更改的模块加入到系统中时就不需要进行另外的训练或学习。

如果说DataFlex屏幕显示是“活动的”，这就意味着在发出Flex-键的任何时候都能执行数据输入和文件维护功能。

例如：如果一个特定的屏幕包含二十个显示窗口，但仅有头五个数据元是要初始输入的，然后在第五个数据项被输入后就可以发出一个SAVE命令。这就不必在每个空白字段后都击RETURN键才退出屏幕，这个特点由于实际减少了必要的击键次数而可以改善数据输入效率。

1.3.4 FLEX—键摘要

命令	动作
ESCAPE	终止程序，返回菜单。
CLEAR(清除)	清除所有的屏幕窗口。
RETURN(回车)	终止光标当前所在的窗口的输入并使光标移到下一个窗口的起始位置。
BACK WINDOW	将光标从当前窗口移回到屏幕上的前一个显示窗口。
(回退 窗口)	
FIND(查找)	查找并显示出与光标当前所在的窗口中的关键码有关的记录。如果窗口中没有数据则对当前窗口有关的字段查找并显示出索引中的第一个关键码的记录。
SUPERFIND (超级查找)	当屏幕显示来自多个数据库文件时，该命令将根据当前光标所在窗口的某个关键码查找记录，然后将查找到并显示出来的与关键码记录相关的记录填入屏幕上的所有数据窗口。
SAVE(保存)	将显示在屏幕上的信息存贮到盘上。如果屏幕包含有一个以上文件的数据，则数据将被写到程序所规定的文件中。
DELETE (删除)	从文件中消除显示的记录并从相应的关键码列表中消除它的关键码。一个记录在它被删除之前必须要查找到。
PREVIOUS RECORD (上个 记录)	查找其关键码恰在光标当前所在窗口的关键码之前的记录。
NEXT RECORD (下个 记录)	查找其关键码紧跟在光标当前所在窗口的关键码之后的记录。
PRINT SCREEN (打印屏幕)	将整个屏幕显示输出到打印机上。