
前言

《程序员指南：字符用户界面(FMLI 和 ETI)》是针对这样的应用程序开发者的：他们想在运行 UNIX System V R4.0 的 ASCII 字符终端上开发一个基于菜单和表格的界面。已有的应用程序可被改写以适应某字符用户界面前端程序，而新的应用程序从设计开始就可以得益于 FMLI 和 ETI 的屏幕管理功能。

FMLI 是一个高层的程序员界面，用以创建符合精心定义的视觉原则的菜单、表格及正文画面。一个应用程序开发者，如果在文件中已经定义了画面(菜单、表格及正文)，实现他们的显示和用户的交互作用就不用再编程了。这种类 shell 的语言由一个解释器处理并且允许开发者规定菜单和表格的布置。FMLI 允许应用程序开发者又快又方便地定制特定的应用程序而不必写 C 语言代码。

ETI 是一个有关屏幕管理的库函数集合(建造在 `curses` 上)，用以加速处理窗口、面板、菜单以及表格的应用程序的开发。ETI 也包含了定义求助、出错及其及类型消息的函数，还包括一些用以又快又方便地显示以及改变这些消息的函数。它是一个用来创建用户界面及应用程序的 C 语言工具箱。ETI 允许开发者去设计一个唯一的或定制的用户界面。

这本指南将告诉您如何使用表格和菜单语言解释器(FMLI)及扩展的终端界面(ETI)两个软件开发工具为您的应用程序编写这样的用户界面。本书假设读者已具有 UNIX System V 操作系统、shell 编程以及/或者 C 语言编程方面的知识。

注意：本指南不打算介绍 UNIX System V R4.0. UNIX 系统 shell 编程或者 C 语言编程。有关 shell 及 C 语言编程的介绍，见前言后“有关文档和培训教程”一节。有关 UNIX System V R4.0 的概述，见《产品概览和总索引》的“产品概述”小节。

本册指南包括独立的两种文档资料：

- 《表格和菜单语言解释器(FMLI)程序员指南》

这本《表格和菜单语言解释器(FMLI)程序员指南》描述了表格和菜单语言，并且告诉您如何应用它为您的用户界面写表格、菜单以及正文画面的说明。它告诉您如何最好地发挥 FMLI 提供的屏幕管理的功能，并且告诉您如何裁剪用 FMLI 写的用户界面的默认显示及功能。

- 《扩展的终端界面(ETI)程序员指南》

这本《扩展的终端界面(ETI)程序员指南》描述了这个扩展的终端界面，并且告诉您如何使用它在 UNIX 系统上写屏幕管理程序。它告诉您如何使用高层的库函数来构建面板、菜单以及表格，它也描述了这些例程是如何与低层的 `curses` 例程以及 `terminfo` 数据库相关联的。

0.1 有关文档及培训课程

有关 AT&T UNIX System V R4.0 的全部书目，请见《产品概览和总索引》一书，其中选择号 307-300 代表此书。下面列出的文档对您学习 FMLI 和 ETI 有帮助。

0.1.1 有关字符用户界面的其它书目

- 《AT&T字符用户界面风格指南》选择号307-096。这本指南为以有“视感”一致性的 FMLI 和 ETIP 创建的用户界面提供了风格准则。
- 《AT&T ETIP设计者用户指南》，pec码1041508，com码105592471。
这本指南解释了如何用ETIP设计程序为应用程序开发一个基于字符的用户界面。
- 《AT&T AOE用户界面程序员指南》，选择号350-302。
这本指南为应用程序开发者，如ISV和VAR，提供了有关字符和图形用户界面工具的一个概述和说明，这些工具用在 AT&T 的“应用操作环境(AOE)”中。

0.1.2 有关 UNIX 系统，shell 及 C 编程的书目

- 《AT&T UNIX System V用户指南》，选择号307-301。这本指南是针对一般用户的有关 UNIX System V R4.0 的介绍。它讨论了操作系统、文件和文件系统、正文编辑、FACE 用户界面，电子邮件所基于的原则，以及 shell 编程的基础知识。
- 《AT&T UNIX System V用户参考手册》，选择号307-302。这本手册描述了 UNIX System V R4.0 的基本命令，这些命令或由用户直接使用，或者通过 UNIX 系统 shell 过程来调用。
- 《AT&T UNIX System V程序员参考手册》，选择号307-314。这本手册描述了支持 C 以及其它编程语言的 UNIX System V R4.0 的编程特征。它包括了 UNIX 系统命令、系统调用、子程、标准 I/O 库例程以及文件格式的说明。

0.1.3 如何订购其它参考书

若要购书请拨下面的号码：

- 在美国大陆：1-800-432-6600
- 在加拿大：1-800-256-1242
- 在欧洲，通过拨+441567 7711从AT&T UNIX 欧洲有限公司购书。（“+”代表国际区号，如伦敦，国际区号为 010。）
- 美国大陆以外：
电话：呼 001-1-317-352-8556

FAX: 传 001-1-317-352-8484或

传 001-1-317-352-8626

Telex: 呼 5101009077

0.1.4 培训课程

有关 AT&T UNIX System V R4.0 的全部培训课程表, 见下列书目:

《AT&T 教育及培训课程目录》

《AT&T UNIX System V 和 C 语言培训》(小册子)

若想得到有关培训课程方面的信息, 请呼下面电话: 1-800-554-6400

0.2 如何对本指南提出意见

我们想为您提供有关 UNIX System V R4.0 的尽可能好的文档。如果您能为我们提出如何改进此指南的建议, 请将标题页后的卡片填好寄给我们。



有关本指南

I.1 简介

本指南介绍 AT&T“表格和菜单语言解释器”(FMLI) R4.0。(要了解如何用 FMLI 命令 `release` 来确认在您的机器上装的 FMLI 是什么版本的方法,请参阅第一章中的“使用 FMLI 应用程序”。)FMLI 是一个为字符用户界面提供创建菜单、表格以及正文画面的高层语言的开发工具。它能产生并协调各屏幕元素,并自动为终端显示提供一种一致性的直观风格。它也为应用程序开发者提供一个直观的、强有力的以及一致性的界面,通过该界面,用户可以访问大量不同的应用程序。

在许多 FMLI 的直观元素被预定义的同时,应用程序开发者有很大的自由度来定义正文元素,并结合可用的直观元素来创建应用程序。

I.1.1 本指南的读者

本指南是为这样的应用程序开发者而编写的:通过一个在仅能使用 ASCII 字符的显示设备上执行的界面,他们或者有兴趣为 UNIX System V 改进一个已有的程序,或者有兴趣开发一个新的程序。

本指南假设读者们具备了 UNIX 系统以及 shell 编程知识(有关这方面的详细信息,请参见《UNIX 系统 V 第 4 版用户指南》。)

I.1.2 本指南的组织

第一章“FMLI 引论”,给出了表格和菜单语言解释器的高层概述以及基本概念。在本指南中使用的“FMLI 应用程序”,指您自己写的应用程序,它们由下面两部分组成:一是由您提供或编写的、用以定义特定功能的 shell 脚本以及其它的程序,一是由系统为用户提供的用以定义菜单、表格以及正文画面的 FMLI 脚本。

为了使您能为您的用户编写足够的 FMLI 应用程序文档,本章对有关 FMLI 应用程序的基本“视感”原则进行了讨论。“如何使用一个 FMLI 应用程序”这一节,概括了在本 FMLI 环境(“视”观)下产生菜单、表格以及正文画面的方法,以及其基本的游动;概括了命令执行的方式,命名键的默认值的指定功能,以及在一个键盘(“感”观)上不存在命名键或者命名键不起作用的情况下,使用替代的击键的方法。

第二章“表格和菜单语言”概括了所有的语言元素,包括语法规则、编程约定、变量的应用、描述字、FMLI 命令、FMLI 内部实用程序、条件语句、中断信号处理以及终端显示属性的应用。

第三章“画面定义文件”描述了菜单、表格以及正文定义文件的构造，解释了在其中可用的 FMLI 描述字的意义，还提供了每一种画面的例子。

第四章“应用层定义文件”，告诉您如何去写一个初始化文件、命令文件以及别名文件，解释了各文件中可用的 FMLI 描述字，并且给出了例子。

附录 A“编程建议及已知的问题”，包括了方便您使用 FMLI 的一些重要信息。开发者应该在设计应用程序之前先阅读本部分。

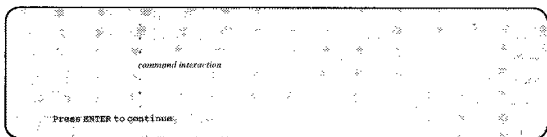
附录 B“命名的键及替代的击键”，综合了命名的键的指定功能，并且为不具备全部可用命名键集合的用户提供了替代的击键。

附录 C“FMLI 手册页”，包括了 FMLI 内部实用程序的手册页、`vsig` 可执行命令以及 `fmli` 命令。

I.1.3 本指南使用的印刷符号约定

本指南中应用了以下印刷符号约定：

- 正文中以大写的 TRUE 和 FALSE 来表示逻辑值“真”和“假”。布尔描述字必须赋以 TRUE 或 FALSE 值，这里：
 - FALSE 表示“假”的字面意义，忽略的情况或者一个非零的返回码。
 - TRUE 表示 FALSE 定义以外的意义。
- 计算机输入输出的字面元素，包括用户输入、程序码、UNIX 系统命令名、FMLI 命令及内部实用程序名，以及其它表格和菜单语言元素都以等宽字体来表示。
- 命令行的替换元素以及表格和菜单语言元素的替换元素都以斜体字表示。
- 屏幕显示的注释(它不是计算机输出的正文，而是作者为读者提供的旁注)以斜体字表示，如下面示例。



- 命令的键以一个硬键形式表示。这包括如 `[ESCAPE]` 或 `[DEL]` 之类的键，以及从 `[F1]` 至 `[F8]` 的功能键。
- 对不能使用命名键的用户，替代的击键的序列也以硬键形式表示。例如，命令键 `[DEL]` 的替代的击键序列是 `[CTRL-X]`。这意味着用户在按住 `[CTRL]`

的同时按下 **[a]**。

较长的替代的击键的序列以硬键序列来表示。例如 **[CTRL-f][3]** 就是在用户没有命令键 **[F3]** (或者 **[F3]** 不起作用) 时必须按键的序列。在这种情况下, 用户必须在按住 **[CTRL]** 的同时, 按下 **[f]** 键, 接着按下 **[3]** 键。

- 标记屏幕的功能键亦以硬键形式表示。例如, 当一个菜单是屏幕上活跃的画面时, 功能键 **[F1]** 有屏幕标号 **[HELP]**。
- 随着键盘的不同, 回车换行键可以被称为 **[ENTER]**、**[RETURN]** 或别的什么。在本指南中以 **[ENTER]** 来代表回车换行键。然而, 若一个键盘仅有 **[RETURN]** 键, 就用它或者 **[CTRL-m]** 来代替。
- 在描述命令句法时, 使用下面的符号约定 (特别是在附录 C 的 FMLI 手册页中):
 - 1 一个命令 (包括命令名本身) 的字面元素以等宽字体来显示
 - 2 命令的替换参数以斜体表示。
 - 3 包围参数的方括号表示此参数为选项。
 - 4 省略号 (...) 用以表示前面的参数可被重复使用。



第一章 FMLI 引论

1.1 简介

本章一般性地描述表格和菜单语言解释器。本章包括四个主要部分：

- “什么是FMLI?”简略地概述了您能用FMLI做些什么，描述FMLI应用程序运行时的屏幕配置，并且解释每一屏幕区域的功能，还定义了画面这个概念。
- “用FMLI编程”描述了三种类型的画面——菜单、表格以及正文画面——您的应用程序就是在其中提供给用户的，并且描述了在您所写的画面定义文件中裁剪画面的呈现的几种方法。在其中也描述了您可以大体上定义您的应用程序特征的应用层文件。
- “应用程序示例”提供了一个简单的FMLI应用程序的例子，并且告诉您如何去执行它以及如何从中退出。
- “使用FMLI应用程序”解释了如何在FMLI应用程序中漫游，如何执行命令(包括UNIX系统命令)，如何使用屏幕标了号的功能键以及如何获得帮助。

1.2 什么是 FMLI?

“表格与菜单语言解释器”为开发者用菜单及表格编写应用程序和应用程序界面提供了一种画面结构，它为您控制着屏幕管理的许多方面。这意味着您不必再操心一些低层的细节，诸如创建和安置画面、为用户提供在画面内或画面间漫游的手段以及处理表格和菜单的应用之类的事，也不必担心您的应用程序将来能够在什么类型的终端上才能运行。FMLI已为您分担了所有这些事情。

FMLI是由两个主要部分组成的高层编程工具：

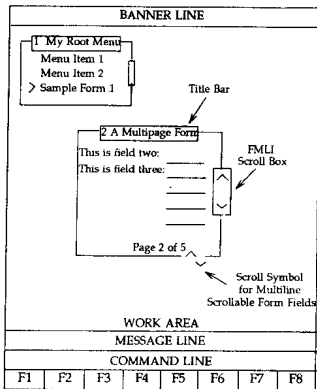
- “表格和菜单语言”即为编写定义一个应用程序如何提供给用户的脚本时所采用的一种编程语言。“表格和菜单语言”与UNIX系统shell编程语言有十分相似的句法，并且还包括下列元素：变量设置及赋值、内部命令和内部函数、特殊字符的转义、输入输出改向、条件语句、中断信号处理以及设置不同终端属性的能力。“表格和菜单语言”也包括“描述字”的集合，这些描述字用来定义和裁剪画面的属性以及您的应用程序的其它一些特征。
- “表格和菜单语言解释器”`fml(1)`，这是一个命令解释器，被用来在一个终端上设置及控制视屏显示，以及使用您程序脚本中的指令来增补FMLI预先定义了的屏幕控制机制。无论是在后台还是在全屏幕方式，FMLI脚本都能调用UNIX系统命令和可执行程序。“表格和菜单语言解释器”类似于UNIX命令解释器`sh(1)`那样进行操作。运行时，它会扫描您所写的程序脚本，因而为您提

供快速构造原型以及易于维护的好处。

1-2-1 屏幕布局

下图表示 FMLI 应用程序运行时的屏幕配置。

图 1-1: FMLI 屏幕



FMLI 将屏幕划分为以下五个区域:

标题行 标题行在屏幕最上面显示一行标题。当FMLI忙时, 它的默认方式是显示一个“正在工作”的图标。然而, 您可以在初始化文件中重新定义标题行。

工作区 工作区是画面显示的屏幕区域(一个画面是由边界包围的独立可滚动的屏幕区域。画面将在下一节“画面”中定义)。这个区域从屏幕的第二行始至倒数第三行止。

小心：如果终端支持硬件功能键，那么FMLI将用屏幕最后一行来显示功能键标号。开发者须认识到这点，因为这样工作区要减少一行（即它将在倒数第四行结束）来为这些标号腾出空间。

消息行 从屏幕底倒数第二行是消息行。由FMLI产生的或者从您程序脚本中产生的消息在这儿显示。默认的方式是消息行消息一直保持到按下下一个键为止；当然，您也可以定义一条消息，让它一直保持显示，或者一直保持到其它的用户动作发生（例如向另一画面游动）。

命令行 命令行处于屏幕倒数第一行。用户通过按`CTRL-]`或`CTRL-]` `[c]` 键来访问它，此时在此行直接出现一个`-->`提示。任何 FMLI 命令、针对应用程序的命令或者 UNIX System V 可执行命令都能从命令行执行。

功能键的屏幕标号 (SLK)

屏幕最后一行显示代表许多键盘上可见的八个功能键 (SLK) 的屏幕标号。SLK 允许用户按下其中一个键来方便地执行功能操作。FMLI 为功能键提供了两组屏幕标号集合。您的程序脚本可以选择让其中一个集合在任何指定时间内显示。FMLI 在第一个集合中预定义 SLK，根据当时起作用的画面类型为每一个 SLK 指定一个默认的屏幕标号及功能。替代的集合不是预定义的——您可以特意为您的应用程序定义它们。在第一个集合中，您可以将功能键 `F1` 到 `F7` 改名或禁用（但不能重定义），可以将功能键 `F8` 重定义。在第二个集合中，您可以为从 `F9` 到 `F16` 的功能键定义针对应用程序的命令。记住，若您重定义了 `F8` 或 `F16`，而又不是定义为 `CHG-KEYS`，则您的用户将会失去访问功能键的替代集合的权力。

既然一些键盘并没有功能键，FMLI就预定义了相当于功能键的“替代的击键”（默认的方式是：在一些终端上，FMLI 试图将替代的击键的定义下行装入到功能键中。若要获取有关这步操作是如何实现和为什么进行，以及您如何禁止这种方式的信息，请参见附录 A“编程建议及已知的问题”。）

1.2.2 画面

一个画面是由边界包围的工作区中的一个独立的可滚动部分。在默认时，画面的维数由 FMLI 决定，并且画面置于工作区内，以便把与其它画面的覆盖部分减至最小。可以同时在工作区中打开几个画面；然而，在任何指定的时刻，只可以有一个当前画面。

当前画面是用户正在其中工作的画面。它通过边界（及边界上的所有特征）的不同来区别于工作区内的其它画面：它的边界以全亮视觉属性来显示，而非当前画面是以半亮视觉来显示的。（在没有半亮视觉属性的终端上，非当前画面显示不同；例如它们中可能是以相反视觉来显示的。）用户的输入应用于当前画面。当前画面可能覆盖工作区内的其它画面。

所有的菜单、表格以及正文画面能显示以下特征：

标题条 每一个画面在它的顶层显示一个标题条。标题条包括由FMLI赋给画面的画面ID号以及画面标题(或是FMLI的默认标题,或是您自定义的)。

滚动盒 每一个显示三行以上信息的画面都在其右边界上显示一个滚动盒。一个滚动盒包括向上符(^)和向下符(v)。若出现了滚动符号,则指示用户:在当前显示在画面内的信息之前(^)和之后(v)还有更多的信息。例如当用户正在看一个只有两页的表格的第二页时,只有向上符(^)将出现在滚动盒内。一个滚动盒仅在画面包含三行以上信息时才出现。当画面内的信息当前都可见时,滚动盒内将是空白。

在底界上的滚动符号

如果当前字段被定义为多行结构且可滚动,那么滚动符号也将出现在一个表格画面的右下边界上。

1.3 用 FMLI 编程

一般地,FMLI 应用程序的程序脚本包括一组画面定义文件的集合,每一个文件定义一个菜单、表格或正文画面。这些文件定义用户执行 FMLI 应用程序所能看见的画面、以及画面内能进行的操作。另外,大多数 FMLI 应用程序都包括三个(可任选)应用层定义文件:一个初始化文件、一个命令文件和一个别名文件。这些文件定义您应用程序的全局特征,例如不同屏幕元素的颜色、针对应用程序的命令以及能使您的代码易读易维护的路径别名。

1.3.1 画面定义文件

画面定义文件是您用表格和菜单语言在其中定义一个画面的 ASCII 文件。可用 FMLI 定义三种类型的画面:菜单画面、表格画面以及正文画面。FMLI 根据画面定义文件的内容以及某些文件的命名约定来识别您所定义的画面类型。下面几节简略地描述这三种类型的画面。如何去编写画面定义文件的详细的解释请见第三章。

1.3.1.1 菜单画面

FMLI 中的菜单是在画面中显示出一个选择表,然后确定用户的选择并且根据选择采取相应的行动的一种方法。菜单画面的标题条显示菜单(Menu)的一个默认名字,或者一个您自己定义的名字,以及由 FMLI 赋给画面的标识号。在默认时,FMLI 在左部对齐的一列中为菜单提供 10 个或者更少的选择项。当选择项数目大于 10 时,FMLI 采用试图将列数保持为行数的三倍的算法,自动地创建一个系列的菜单。菜单中项的数目决定了每一列的高度及列数。然而,您也可以在菜单中显式地定义您想要的行数和列数。

如果对整个菜单来说,项数过多,不能在屏幕上一次显示完的话,菜单将被设为单列的菜单,并且可以滚动(有关描述 FMLI 计算菜单中行数和列数方法的表,请参见附录 A“编程建议及已知的问题”)。如果在画面右边界的滚动盒内出现某个滚动符(^表示向上,

v 表示向下), 那么除其它菜单游动键以外, 命名键 **SCROLL-UP** 和 **SCROLL-DOWN**) 也起作用。如果菜单可滚动, 那么有许多菜单游动键支持环绕(游动键如何作用的全部描述请见附录 B)。

单一选择和多重选择的菜单

您可以将菜单定义为单一选择的或多重选择的。在单一选择菜单中, 用户仅能选择一项。当用户按下 **ENTER** 键时, 与光标位置上的项相关的反引号表达式及 FMLI 命令分别立即求值或执行。

在多重选择菜单中, 用户可以“标记”不止一个的选择。一个被标记的菜单项以项左边的一个星字符(*)指示。在多重选择菜单中, 用户按下 **MARK** 来标记某项, 立即求值与此项相关的表达式; 然而忽略与被标记项相关的 FMLI 命令。在标记完所有您想标记的项之后, 用户按下 **ENTER** 键, 给 **done** 描述字赋值(第二章“表格和菜单语言”中讨论了表达式、描述字以及 FMLI 命令)。

1.3.1.2 表格画面

表格是显示和提示信息的方法。表格的标题条显示表格的默认名(Form), 或者一个您定义的名字以及由 FMLI 提供给画面的画面 ID 号。表格由字段组成, 每个字段有两部分: 一个是字段标号(字段名), 一个是将键入字段值的区域。您可以验证用户在字段里键入的值: 若您为字段定义了验证测试, 则在键入值通过验证测试之前, 用户就不能游动到另一字段去, 也不能存取表格。您可以将字段输入区域定义为多行的和/或可滚动的。若可滚动, 则当光标在多行字段上定位时, 滚动符号将出现在画面底边界的右侧; 如果字段是单行字符, 那么滚动符号将出现在字段输入区域中。对用户而言, 一个表格看起来就像空格填入式调查表。关于能在表格画面内游动以及编辑表格字段的游动键的描述, 请见附录 B“命名键及替代的击键”。

多页表格

一个表格可有不止一页的长度。若多于一页的话, 当表格画面第一次打开时显示第一页。在滚动盒内的向上向下符是 FMLI 用以通知用户他们正处于第一页、最后一页或者表格中间某页的一种手段。如果您想给出用户看到的页的更为精确的信息, 那么您必须在表格画面定义文件中定义相应的信息。第三章“表格定义文件”中将解释如何去包含诸如 5 页长度表格的第 2 页这样的信息。

选择菜单

“选择”菜单是向用户显示表格内某字段的有效选择的方法。当您为一个字符定义了一个“选择”菜单时, 您可以选定用户是否将触发字段内的所有选择, 或者各选择是否显示为一个弹出式菜单。每一种方式中, “选择”菜单都由 FMLI 命令 **Choices** 来访问(在默认方式下由功能键 **F2** 来表示)。

当用户从一个“选择”菜单内选好一个值时，它被自动地键入“选择”菜单所对应的字段中去。在附录 B 中描述的菜单游动键可用于一个“选择”菜单，但用户须注意，如果他们从“选择”菜单内游动出来，那么“选择”菜单将立即消失。

1.3.1.3 正文画面

正文画面主要用来显示只读信息，例如对于用户的联机帮助。正文画面的标题条显示画面的默认名(Text)或者您自己定义的名字以及由 FMLI 赋给画面的画面 ID 号。如果滚动符(^)或(v)的某一个出现在画面右边界的滚动盒内，那么除其它正文画面游动键外，SCROLL-UP和SCROLL-DOWN键将起作用。如果您定义的所有正文不能装入画面，那么滚动将自动进行。您可以将正文画面定义为“可编辑的”，即用户可以使用正文画面编辑键来改变显示的正文(参见附录 B)。

1.3.2 应用层定义文件

应用层定义文件在总体上定义应用程序的属性。有三个可选的应用层文件：初始化文件和命令文件，它们允许您裁剪应用程序的呈现及功能；别名文件允许您对代码内的源文件进行流水线式引用。下面小节简略地描述了这三个应用层定义文件。有关如何编写的详细的解释请见第四章。

1.3.2.1 初始化文件

初始化文件总体上定义应用程序的属性。您能定义一个简介性的画面(如版权声明)、对默认标题行的改变值、FMLI 屏幕上不同元素的色彩、用户是否可以直接从您的应用程序访问 UNIX 系统以及指定给标记屏幕的键的显示名字和命令，等等。

1.3.2.2 命令文件

一个命令文件允许您为应用程序的用户定义新的命令，重定义或禁用已存的 FMLI 命令。命令文件的内容将反映在 FMLI 命令菜单(将在本章下一节讨论)中，其中定义的命令也能在 FMLI 命令行上由应用程序的用户执行。

1.3.2.3 别名文件

别名文件包括形如 *alias=pathname* 的行。可以将一个文件或者设备的单个路径指定给一个别名，也可以将要搜索(类似于 UNIX shell 中的 *\$PATH* 的搜索方式)的一系列路径指定给一个别名。使用别名将使您的其它定义文件内的代码更易读。

1.3.3 终端独立性

FMLI R4.0 使用 UNIX System V R4.0 terminfo 数据库来确定与终端有关的功能的值。若环境变量 `TERMINFO` 没有置值，则数据库的默认路径是

`/usr/share/lib/terminfo`。可将未在数据库中描述的新的终端加到 `terminfo` 数据库中，以新终端名字的第一个字符命名的相应子目录下。例如，5425 终端的说明将是 `$TERMINFO/5/5425`。

为了保证终端针对您的 FMLI 应用程序正确地初始化，在调用您应用程序的可执行程序或程序脚本中包括命令

```
tput init
```

若您不打算这样做，那么您在应用程序的文档中必须提醒用户在 `TERM` 变量赋值并且移出后将这个命令加进他们的 `.profile` 文件中。

注意：当用户在使用完一个通过 FMLI 运行内部实用程序来执行的全屏幕的应用程序后，返回到 FMLI 应用程序，这时可能丢失终端属性的设置值。为防止这种情况发生，在返回到 FMLI 之前，全屏幕的应用程序应执行 `tput init` 命令。

FMLI 可在满足下列条件的任意异步终端上工作：

- 每行显示80个字符
- 至少具有22条同时可见行
- 在主机上有正确的 `terminfo` 入口。

如果您定义画面的大小以及位置以适合屏幕的限制，那么 FMLI 也可能在更小的屏幕上运行。然而，可能会截断诸如功能键的屏幕标记此类的一些屏幕布局元素。

1.3.3.1 异常中止后的恢复

在 FMLI 应用程序异常中止的情况下，用户可以执行

```
CTRL-] stty sane
```

来恢复屏幕。而在这之前，可能不在屏幕上显示用户的输入，而只是显示出计算机被“挂起”或者已下机。如果用户执行了上述命令来恢复，那么也有必要执行

```
CTRL-] stty tab3
```

来保证屏幕正常(否则画面的边界可能会被扭曲)。

注意：在默认时，FMLI 会在一些终端上试图将替代的击键的定义下行装入到功能键中。附录 A “编程建议及已知的问题”，将给出这是如何做的，以及您如何能禁止这样做的信息。

1.4 应用程序示例

这是一个由名为 `Menu.Sample` 的菜单画面定义文件，名字为 `Text.Welcome` 的正画面定义文件组成的 FMLI 应用程序的例子。

图 1-2: Menu.Sample: 一个简单的菜单画面定义文件

```
menu=TOP MENU

name=date
action='date|message'nop

name=welcome
action=open Text.welcome

name=exit
action=exit
```

Menu.Sample 文件是根据为 FMLI 画面定义文件定义的约定而命名的(第二章包括了文件命名约定的完备的讨论), Menu.Sample 定义了当示范应用程序运行时初始打开的画面。

代码的第一行是 FMLI 的一个描述字 **menu**, 它用来定义在菜单画面的标题条上出现的标题。当您不显式地在画面定义文件中定义描述字时, 许多 FMLI 的描述字都有一个默认值。如果 **menu** 描述字不在此文件内定义的话, 菜单的标题将默认为 **Menu**。

以下两行代码定义第一个菜单项。将在菜单中出现的此项的名字由 **name** 描述字定义。**action** 描述字定义了当用户从菜单中选择了 **data** 项时发生的操作: UNIX 系统 **data(1)** 命令将运行, 且输出将由管道引入 FMLI 内部实用程序 **message(1F)**, 此程序再把 **date** 的输出显示在 FMLI 的信息行上。**nop**(no operation 之意)是什么也不做的 FMLI 命令, 但它却必须出现, 因为 FMLI 期待 **action** 描述字最终被赋予一个 FMLI 命令, 若 **nop** 不出现的话, 终端会响铃。

第二项将菜单项名定义为 **welcome**, 选中 **welcome** 时要执行的动作定义为在工作区打开另一个画面——在此情况下是一个正文画面 **Text.welcome**, 它将显示一条欢迎消息。图 1-3 显示了这个正文画面定义文件 **Text.welcome** 的内容。

图 1-3: Text.welcome: 一个正文画面定义文件

```
title="WELCOME"
rows=3
text=" Welcome to my application.
I hope you enjoy yourself
while you are using it."
```

第三个菜单项将 `name` 认定为 `exit`。这个动作是在用户选择此项时去运行 FMLI 命令 `exit`。这将导致 FMLI 会晤期的结束，用户将返回到 UNIX Shell。

文件 `Text.welcome` 是根据关于正文画面定义文件的约定而命名的，并且定义了一个正文画面。`title` 描述字定义了将在正文画面标题条中出现的标题，在这个例子中是 `WELCOME`。`rows` 描述字定义了正文画面的垂直高度（以正文行数来衡量）。此例中的画面的宽度由 FMLI 来决定。（参见附录 A“编程建议及已知的问题”中关于 FMLI 如何计算画面 `rows` 和 `columns` 的讨论。）`text` 描述字定义了将在正文画面体内显示的单词。

为了调用此应用程序，可按如下所示调用 `fmli` 命令：

```
fmli Menu.Sample
```

此例中，`Menu.Sample` 规定为打开的初始画面。初始画面是调用 `fmli` 时打开作为 `fmli` 的实参的画面。初始画面可以不止一个，并且在其应用程序运行的全过程中，所有的初始画面都在工作区里显示（即，用户不能取消、或关闭一个初始画面）。

为了从这个 FMLI 应用程序中退出，用户可以从 TOP MENU 中选择 `exit`，因为 FMLI 命令 `exit` 已被定义为一个菜单项了。从 FMLI 应用程序中退出的一种替代的方法是按 `CTRL-3`（或者 `CTRL-[C]`）来访问 FMLI 命令行，然后键入 `exit`。

在下一小节中，这个示例应用程序将被用来说明一个 FMLI 应用程序对用户而言是什么样子的。如果您想对这个采用一个依次传递使用的例子的讨论加深一些印象，那么您可以在您的文件系统中备一份 `Menu.Sample` 和 `Text.welcome` 的副本，并且如上所述执行 `fmli` 命令。

1.5 使用 FMLI 应用程序

这一节讨论 FMLI 应用程序的“观感”方式。它包括了提供菜单、表格和正文画面，以及提供的 FMLI 屏幕环境的其它可视元素（“视”观）、游动的基本方式，命令执行的方式，赋予命名键的默认功能，以及在命名键不起作用或键盘上没有相应命名键情况下如何使用替代的右键的序列（“感”观）。

因为下列两个理由使得这些信息很重要。首先，使得您能理解那些 FMLI 应用程序中“随之而来”的特征；因而可让您设计应用程序利用这些特征。第二，可以对那些需要包括在您的用户文档中的关于使用应用程序的信息进行总结。我们不能帮您描述在您的菜单、表格以及正文画面中有什么，但是我们可以帮您去描述那些对任何 FMLI 应用程序都必须相同的特征：在画面之内和之间游动、进行菜单选择、执行命令以及申请帮助。

当您执行命令 `fmli Menu.Sample`（上面刚刚讨论过 `Menu.sample` 的 FMLI 脚本）时，终端屏幕看起来会是这样的：