

2000系列
软件资料

VAX FMS
表格驱动程序参考手册

2000系列资料出版中心



前 言

美国DEC公司的VAX—11系列机是举世公认的优秀的32位超级小型机系列。它不但具有功能完善的指令系统、灵活巧妙的寻址方式、多种数据类型以及虚拟存贮等特点，并且其结构面向操作系统，具有丰富的软件支持。

自七十年代末以来，VAX 系列机已在世界各得到广泛应用，我国已有不少单位引进该系列的各种机型，它具有广阔的发展前景。为推动国内计算机事业的发展，并考虑到 VAX 用户及各高等院校教学的实际需要，主管部门组织电子工业部华北计算所、中国科学院沈阳计算所、中国科学院高能物理所、成都电讯工程学院、暨南大学、北京信息工程学院、航天工业部一院十二所等单位，成立了VAX系列机资料出版中心，组织经验丰富的软件专业人员 翻译出版全套的VAX随机软件资料。

这套资料的第一批包括VAX/VMS (3.6版) 十卷38册 (VAX/VMS的一般介绍、命令语言和系统信息、文本编辑和格式化程序、程序开发工具、系统服务和 I/O、运行时间库、VAX—11记录管理服务、兼容方式、系统程序设计、系统管理及操作)。网络一卷1册、可选的VAX/VMS选件八卷25册 (包括：FORTRAN、BASIC、PL/I、COBOL、BLISS—32、C、PASCAL、CORAL—66等语言)、DBMS (数据库) 一卷12册，CDD (公共数据字典) 一卷3册，数据检索一卷6册，~~共三十二卷~~85册。现已全部出版。

第二批资料包括VAX Rdb/VMS ~~(七册)~~、VAX—11 LISP语言 (三册)、VAX—11 CEP/VMS (五册)、VAX Cluster ~~(六册)~~、VAX—11 RGL (四册)、Micro VAX (五册)，共二十五册。

本套第三批资料包括VAX RMS ~~(七册)~~、VAX TMS (八册) 和 VAX GKS (四册) 三卷共十八册。

迄今为止，这是一套最完整、最系统、最丰富、最实用的软件资料。这套资料对VMS系列各档机的用户是必读资料：对从事计算机研制的各单位以及高等院校计算机工程系的教学是重要的参考资料：对各大专院校，各省、市图书馆也是珍贵的馆藏资料。

《VAX FMS表格驱动程序参考手册》(序号：AA—L319B—TE) 操作系统及版本：VAX/VMS V4.0，软件版本：VAX FMS V2.2是由电子工业部华北计算所黄仲孚、张金荣翻译，由《小型微型计算机系统》编辑部编辑、出版、发行。

由于时间仓促、水平有限，因此一定有不少错误和不妥之处，敬请广大读者批评指正。

14586

2000系列资料出版中心

一九八八年一月

序 言

本手册描述 VAX FMS 表格驱动程序, 在VAX/VMS操作系统中使用的表格管理系统的运行时成分。

读者对象

本手册针对程序员来写的, 他们希望把FMS 与用BASIC, BLISS_32, C, COBOL, FORTRAN, PASCAL或PL/I书写的任何他们的应用程序一起使用 (这些语言在《VAX FMS语言接口手册》中叙述)。读者不仅要熟悉程序设计语言, 还要熟悉 VAX/VMS系统。

还希望读者熟悉《VAX FMS实用程序参考手册》第2章表格特征中的信息。此材料讨论了在设计表格时通过表格编辑程序或表格语言指定的表格特征。

要求缺乏FMS经验的读者在阅读本手册之前读一下《VAX FMS介绍》

各章概括

第1章给出了表格驱动程序的综述, 扼要讨论了12个基本方面。

第2章讨论了表格驱动程序与表格描述、用户动作例程、终端操作员、键功能和键代码、调用状态和异步系统内中断 (ASI) 等的交互作用。

第3章提出了程序设计技巧和取自范例应用程序的用各种语言书写的例子。FMS 范例应用程序 (SAMP.BAS) 是FMS发放解质的一个部分, 设计为解释程序和学习工具。

(从SAMP取例子出现在整个FMS资料中。)

第4章展示了怎样把目的模块与表格驱动程序 (还可能有) 内存留驻的表格或用户动作例程联结在一起。本章还讨论FMS程序中终端的使用。

第5章以字母顺序描述所有的表格驱动程序调用, 每个首先给出调用和它的参数的一般格式, 后随以所有参数的定义, 以它们必须被指定的顺序给出 (以及有关怎样被传递和它们是否可读或写的信息), 后面是描述调用做了什么, 最后是作为调用执行结果可能返回给程序的一系列状态代码 (以字母顺序)。

整本手册中调用的名字通常为CDISP。虽然在程序中实际发调用时, 所有 FMS 调用的名字均用前缀FDV\$开头 (例如, FDV\$CDISP)。为了找出发调用的专门于语言的方法, 你必须参考《VAX FMS语言接口手册》中相应章节。

本手册中, 片语“GET类调用”涉及任何下列调用: GET, GETAF, GETAL 和 GETSC (但不包括GETDL)。片语“PUT类调用”涉及调用PUT, PUTAL, PUTD, PUTDA和PUTSC (但不包括PUTL)。

附录A给出了对所有FMS调用信息的概括。

资料约定

大写字母 在命令和例子中指出用户要打入的项要与所示的完全一样。

小写字母 在命令和例子中指出用户要用字或值来替代的变量。

方括号 [] 指出此项是选用的

花括号 {} 在它括起来的一列元素中要选择一个。用直条来分开各种选择。

CTRL/X 指用户按下标记为CTRL的键并压另一个键。

GoldX 指用户在压第二个键之前, 先压Gold键。

... 指出前面的项可被重复。

目 录

前言

第一章 引论

1.1 终端, 工作空间, 表格和字段	1
1.1.1 终端	1
1.1.2 工作空间	2
1.1.3 表格	2
1.1.4 字段	2
1.2 终端控制区和表格工作空间	2
1.3 表格管理调用	3
1.3.1 控制调用	3
1.3.2 表格级调用	4
1.3.3 字段级调用	4
1.3.4 实用程序调用	4
1.4 内存驻留表格和表格库	5
1.5 多终端和多表格操作	5
1.6 调试方式	5
1.7 卷动操作	6
1.8 用户动作例程	6
1.9 命名的数据	6
1.10 终端键功能	6
1.11 当前状态	6
1.12 操作员支持	8
1.12.1 帮助	8
1.12.2 屏幕刷新	8

第二章 表格驱动程序的交互作用

2.1 与表格描述的交互作用	9
2.1.1 存贮和存取表格描述	9
2.1.2 显示表格	10
2.1.3 终端控制	11
2.1.4 利用工作空间存贮表格	12
2.1.5 帮助功能	13
2.1.6 字段处理顺序	13
2.1.7 文本、字段标记字符和显示属性	14
2.1.8 处理字段	14

2.1.8.1 字段图象	14
2.1.8.2 向右对齐和向左对齐字段属性	14
2.1.8.3 清字符和充填字符属性	15
2.1.8.4 缺省字段值	15
2.1.8.5 自动制表属性	15
2.1.8.6 要求响应和必须充填属性	15
2.1.8.7 固定小数点属性	15
2.1.8.8 只显示属性	16
2.1.8.9 不回送属性	16
2.1.8.10 管理程序专用属性	16
2.1.8.11 卷动	16
2.1.8.12 日期和时间属性	16
2.2 用户动作例程	16
2.2.1 字段完成UAR	17
2.2.2 帮助UAR	18
2.2.2.1 帮助前UAR	18
2.2.2.2 帮助后UAR	18
2.2.3 帮助请求处理	19
2.2.4 功能键UAR	20
2.2.5 在UAR中的合法动作	21
2.3 与终端操作员交互作用	22
2.3.1 通知出错和从出错中恢复	22
2.3.1.1 帮助键和帮助消息	23
2.3.1.2 从你的程序检查操作员响应	23
2.3.1.3 刷新屏幕: 打CTRL/R	23
2.3.2 字段编辑功能	23
2.3.2.1 VT100另一种替代的小键盘方式	24
2.3.2.2 在字段中光标的初始位置	24
2.3.2.3 插入字段值: 缺省功能	24
2.3.2.4 带符号的数字图象	24
2.3.2.5 删字符	25
2.3.2.6 删字段	25
2.3.2.7 光标右移	25
2.3.2.8 光标左移	25
2.3.3 转换插入方式	26
2.3.4 字段结束符	26
2.3.5 字段结束符和表格驱动程序调用	28
2.3.6 字段结束功能	29
2.3.6.1 通知表格完成	29
2.3.6.2 把光标挪到下个字段	29

2.3.6.3 把光标挪到上个字段	30
2.3.6.4 往回卷	30
2.3.6.5 向前卷	31
2.3.6.6 往回出卷动区	31
2.3.6.7 向前出卷动区	32
2.3.6.8 非法结束符的交互作用	32
2.3.7 另一种替代的小键盘方式结束符	32
2.4 键功能和键代码	33
2.4.1 表格驱动程序键功能	33
2.4.2 表格驱动程序键代码	33
2.4.2.1 控制键	34
2.4.2.2 换码序列	35
2.4.2.3 Gold序列	36
2.4.3 定义键	40
2.5 检查调用状态	40
2.5.1 应用程序开发用的调试方式支持	42
2.5.2 通知终端操作员关于程序出错	43
2.6 AST考虑	43

第三章 程序设计技巧和例子

3.1 卷动	44
3.1.1 控制卷动区	44
3.1.2 向前卷	45
3.1.3 往回卷	45
3.2 验证一个字符的字段—使用UAR	46
3.3 产生硬拷贝—使用命名的数据	46
3.4 存贮消息文本—使用命名的数据	48
3.5 把功能键转换成字段进入	49
3.6 对功能键的过滤	51
3.7 对字段的范围检查	52
3.8 模拟GETAL调用	54
3.9 压缩表格的显示时间	56
3.10 检查状态—三种方法	57
3.11 调页	59
3.12 FMS先进的程序设计	60
3.12.1 FMS性能	60
3.12.1.1 FMS 库性能	60
3.12.1.2 表格驱动程序性能	60
3.12.2 复盖表格设计	60

第四章 联结应用和建立终端

4.1 联结	64
4.1.1 与表格驱动程序库的联结	64
4.1.2 与内存留驻表格的联结	64
4.1.3 与UAR向量的联结	65
4.2 在FMS程序中终端的使用	65
4.2.1 终端特征	65
4.2.2 直接的终端输出	65
4.2.3 在程序结束时的终端态	66
4.2.4 固件错的绕过工作	66

第五章 表格驱动程序调用

5.1 ADLVA—改变数据行属性	67
5.2 AFCX—改变字段上下文	68
5.3 AFVA—改变字段视频属性	69
5.4 ATERM—连接终端	70
5.5 AWKSP—连接表格工作区	72
5.6 BELL—振终端铃	73
5.7 CANCL—取消调用	73
5.8 CDISP—清除屏幕并显示表格	74
5.9 CLEAR—清除屏幕	75
5.10 CLEAR_VA—清除视频属性	75
5.11 DEL—从常驻内存表格列中取消表格	76
5.12 DFKBD—定义键盘	76
5.13 DISP—显示表格	78
5.14 DISPW—显示装入的表格	79
5.15 DPCOM—把逗号定义为小数点	80
5.16 DTERM—断开终端	81
5.17 DWKSP—断开表格工作区	81
5.18 FIX_SCREEN—恢复终端屏幕的改写行	82
5.19 GET—取指定字段的值	82
5.20 GETAF—取任意字段的值	83
5.21 GETAL—取所有字段的值	84
5.22 GETDL—从终端取数据行	85
5.23 GETSC—取上卷域的当前行	86
5.24 ILTRM—回送非法终止符	87
5.25 LCHAN—对表格库文件置通道	88
5.26 LCLOS—关闭表格库	89
5.27 LEDOF—断开终端LED	89

5.28 LEDON—接通终端LED	90
5.29 LOAD—装入表格而不显示	90
5.30 LOPEN—打开表格库	91
5.31 NDISP—标记当前工作区中的表格为不显示	91
5.32 PFT—处理字段终止符	92
5.33 PUT—输出值到指定字段	94
5.34 PUTAL—输出值到所有字段	96
5.35 PUTD—输出缺省值到指定字段	96
5.36 PUTAD—输出缺省值到所有字段	97
5.37 PUTL—输出行到屏幕	98
5.38 PUTSC—输出数据到滚动域的当前行	98
5.39 READ—读表格到存储器中	99
5.40 RET—指定字段的值	99
5.41 RETAL—对所有字段回送值	100
5.42 RETCX—回送当前上下文	100
5.43 RETDI—回送由索引指定的命名数据	102
5.44 RETDN—回送由名字指定的命名数据	102
5.45 RETFL—回送表格行	103
5.46 RETFN—回送当前字段名	104
5.47 RETFO—按顺序回送字段名	104
5.48 RETLE—回送指定字段的长度	105
5.49 RFRSH—恢复屏幕	105
5.50 SCR_WIDTH—置屏幕宽度	106
5.51 SIGOP—操作员信号	106
5.52 SPADA—置小键盘为应用方式	107
5.53 SPOFF—断开只检查方式	107
5.54 SPON—接通只检查方式	108
5.55 SSIGQ—置信号为无声方式	108
5.56 SSRV—指定状态记录变量	108
5.57 STAT—根据上次调用回送状态	109
5.58 STERM—置当前终端	109
5.59 STIME—置字段输入超时	110
5.60 SWKSP—置当前工作区	110
5.61 TCHAN—置终端通道	111
5.62 USER_REFRESH—建立用户刷新程序	111
5.63 WAIT—等待操作员	112

附录A VAX FMS 表格驱动程序调用

A.1 与VAX语言无关的表示法	114
A.2 表格驱动程序调用的过程参数表示法	114

第一章 引 论

表格驱动程序 (FORM DRIVER)，这个 FMS 的运行时分，是与你的程序相联结的子例程包。表格驱动程序从你的程序接受调用，维护 FMS 数据结构，并发终端 I/O 调用与终端操作员通讯。

如图 1—1 所示，表格驱动程序是 FMS 应用的三个部分之一。

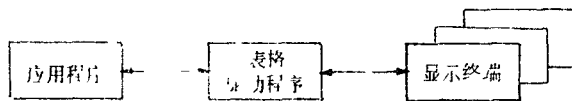


图 1—1: 表格驱动程序通讯

下面12个方面的知识是了解表格驱动程序的基础，本章对这些方面提供扼要的导引：

- 终端，工作空间，表格和字段
- 终端控制区和表格工作空间
- 表格管理调用
- 留驻内存的表格和表格库
- 多终端和多表格操作
- 调试方式
- 卷动操作
- 用户动作例程
- 命名的数据
- 终端键功能
- 当前状态
- 操作员支持

1.1 终端，工作空间，表格和字段

表格驱动的主要工作是对终端映象、表格工作空间、表格和表格字段的加工。

1.1.1 终端

表格驱动程序控制一个或多个终端，实现象显示表格、从终端操作员那儿征求数据和显示消息等任务。终端受控于表格驱动程序发送给它的字符序列和换码序列。

使用 FMS 的应用程序可以处理在任何 VT100/VT200 兼容终端或 VT52 终端上的表格。表格驱动程序支持管理终端的初始化、它们的使用和它们的释放等的 FMS 调用。

在 FMS 应用中，每一终端与一个单独的内存区域相关联。这个区域被称之为终端的控制区或 TCA。一旦 TCA 与终端相关联，应用程序通过发出指定了终端 TCA 的调用来控制该终端的活动。

1.1.2 工作空间

工作空间是内存的区域，表格驱动程序把正被处理的表格的表格描述存在它的里面。与一个终端相关联的可以不止一个表格工作空间，但是每个工作空间同时只能与一个终端相关联。表格驱动程序支持初始化工作空间，把它们与 TCA 相关联，用表格描述装入它们，从 TCA 释放它们等调用。

1.1.3 表格 (FORM)

表格由 (1) 背景文本和字段 (它们被显示在屏幕上)，和 (2) 命名的数据 (它不被显示) 所组成。从内部来看，表格由一些数据结构所组成，表格驱动程序用它们来创建和加工屏幕上的图象。这些数据结构 (被称作表格描述) 由 FMS 实用程序 (表格编辑程序，表格语言翻译程序，和表格升级实用程序) 建立的。(请见《VAX FMS 实用程序参考手册》。)

表格驱动程序支持一些调用，它们将这些表格描述装入工作空间并实现对表格和它们的字段定义的许多功能。当表格被装入工作空间时，工作空间便与此表格相关联。自然，对工作空间的涉及，也就是涉及存在其中的表格。

因为在某一时间工作空间只能一个终端相关联，所以为了将给定的表格在多个终端上显示就需要同样将此表格装入到其它终端相关联的工作空间中。

1.1.4 字段

字段是表格的一个部分，可以有可变的数据与它相关联。表格驱动程序支持多种控制字段数据加工的调用。有些调用允许终端操作员进入或改变字段数据。另一些调用允许应用程序显示字段中的数据，以便取得操作员进入字段中的数据，或者改变字段的显示属性。

在创建表格时赋予的字段属性影响到表格驱动程序处理字段输入和输出的方法。例如，具有要求响应属性的字段需要终端操作员在字段中至少进入一个字符。

当表格驱动程序用单个调用处理一组字段 (在一个表格中) 时，按照表格描述中指定的顺序处理它们。表格驱动程序来控制从字段到字段的移动、字段值返回到你的程序或写到终端上的顺序、以及在表格中“头一个”和“最后一个”字段的定义。

1.2 终端控制区和表格工作空间

终端控制区 (TCA) 和表格工作空间以谱系形式维持，如图 1—2 所示。图示中的应用程序使用了若干终端。附加到应用程序上的是一系列 TCA，它们告诉表格驱动程序它能使用那些终端。附加在第一个 TCA 上的是一系列工作空间，它们可包含在该 TCA 的终端上作处理的表格。类似地，在第二个 TCA 上也附加了一系列工作空间。

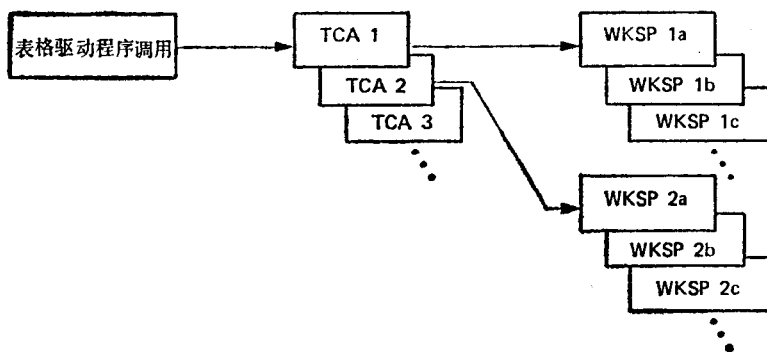


图1—2 终端控制区和工作空间

虽然你是通过名子来涉及表格和字段的，但通常不需要涉及 TCA 名来指出 特定的终端，或者涉及工作空间名来指出你希望将表格装在那儿。表格驱动程序建立“当前终端”和“当前工作空间。”牵涉到终端或者工作空间的每个调用，除非明显地指出另一个 TCA 或工作空间，便作用在当前终端或当前工作空间上。

表格驱动程序提供了一些调用把另外的终端或另外的工作空间作为当前的。通过在你的程序中包含这些调用，就可避免对你的程序中发出的每个调用必须指定终端或工作空间。

1.3 表格管理调用

表格驱动程序支持四类表格管理调用：

1. 控制
2. 表格级
3. 字段级
4. 实用程序

这些调用在下面各节中引入并在第 5 章中作全面的描述。

1.3.1 控制调用

你的程序发下列调用在 FMS 的单位（如终端、表格工作空间、文件和表 格 库）之间建立连接和去掉连接：

ATERM	附加终端
AWKSP	附加表格工作空间
DTerm	去掉附加终端
DWKSP	去掉附加表格工作空间
LCHAN	为表格库文件设置通道
LCLOS	关闭表格库
LOPEN	打开表格库
STERM	设置当前终端
SWKSP	设置当前工作空间
TCHAN	设置终端通道

1.3.2 表格级调用

你的程序发下列调用实现表格级操作:

CDISP	清屏幕并显示表格
DEL	从内存删表格
DISP	显示表格
DISPW	显示装入的表格
GETAL	取所有字段值
LOAD	装入表格但不显示
NDISP	把在当前工作空间中的表格标记为不显示的
PUTAL	把值输出到表格中的所有字段
PUTDA	把缺省值输出到表格中的所有字段中。
READ	把表格读到内存
RETAL	返回表格中所有字段的值

1.3.3 字段级调用

你的程序发出下列调用来实现表格上的字段级操作:

AFCX	改变字段上下文
AFVA	改变字段显示属性
GET	取指定字段的值
GETAF	取任何字段的值
GETSC	取卷动区的当前行
PFT	处理字段结束符
PUT	把值输出到指定字段
PUTD	把缺省值输出到指定字段
PUTSC	把数据输出到卷动区的当前行
RET	返回指定字段的值
RETFN	返回当前字段名
RETFO	依次返回字段名
RETLE	返回指定字段的长度

1.3.4 实用程序调用

你的程序发出下列调用来实现不属于以上各类的操作:

ADLVA	改变数据行显示属性
BELL	终端响铃
CANCL	撤消调用
CLEAR	清屏幕
CLEAR__VA	清屏幕显示属性
DFKBD	定义键盘
DPCOM	把逗号定义成十进小数点

FIX_SCREEN	修复终端屏幕的重写行
GETDL	从终端取数据行
ILTRM	返回非法的结束符
LEDOF	断开终端的 LED
LEDON	接通终端的 LED
PUTL	把行输到屏幕上
RETCX	返回当前的上下文
RETDI	按索引返回命名的数据
RETDN	按名字返回命名的数据
RETFL	返回表格行
RFRSH	刷新屏幕
SCR_WIDTH	置屏幕宽度
SIGOP	信号操作符
SPADA	把小键盘置成应用方式
SPOFF	断开管理员专用方式
SPON	接通管理员专用方式
SSIGQ	把信号置成静止方式
SSRV	指定状态记录变量
STAT	返回来自最后调用中的状态
STIM	设置字段进入超时
USER_REFRESH	建立用户提供的刷新例程
WAIT	等待操作员

1.4 内存留驻的表格和表格库

表格驱动程序从内存留驻的表格描述集合或者从你在表格驱动程序调用中指定的表格库里取得各种调用中所需要的表格。通过把表格建造在应用程序中或者在运行时把表格装入到内存留驻的表格集合里的办法使得表格留驻内存。内存留驻的表格模块通过表格应用支持创建；表格库由表格库管理程序建造和维护。

1.5 多终端和多表格操作

使用表格驱动程序的应用程序可控制一个或多个终端，通过使用控制调用（STERM或SWKSP）来定义新当前终端或当前工作空间的办法在每个终端上加工一个或多个表格。注意在本版本的 FMS 中，你可用这些调用同时只能控制一个终端。

1.6 调试方式

表格驱动程序有操作的调试方式，对使用 FMS 的应用程序的调试有用。当应用程序处于 FMS 的调试方式时，它正常运行直到表格驱动程序返回指出错误的状态时为止。此时，在屏幕的底部行显示消息，表格驱动程序等待操作员压“进入表格”键后再继续。

(任何其它的输入均不予理睬。)

表格驱动程序的调试方式与任何其它的调试手段无关联。

1.7 卷 动 操 作

你可在你的程序中使用 GET、GETAF、GETSC、PFT、TUT和 PUTSC 调用, 取得 VT100 的卷动能力的好处。用这些调用可控制在表格中的卷动区域。

对 VT52终端来说, VT100/VT200的硬件卷动能力是通过软件来模拟的。因此, VT52 卷动操作的性能表现为慢得多。

1.8 用户动作例程

当设计表格时, 表格设计人员可以指出由你自己提供的子程序让表格驱动程序作为表格处理的一个部分来调用。这些子程序被称之为用户动作例程 (UAR)。

用户动作例程在任何下述情况下被调用:

1. 当一个字段的处理完成时;
2. 当终端操作员请求帮助时;
3. 当终端操作员压功能键时;
4. 当请求屏幕刷新操作时。

在对你的程序进行联结时, 必须包括一个目的模块, 它包含要被调用的所有用户动作例程的名字。你使用 FMS/VECTOR 命令来生成这样的目的模块。(见本手册第4章和《VAX FMS实用程序参考手册》第6章表格应用支持。)

1.9 命名的数据

命名的数据是进行表格设计时首先与表格相关联的数据。它不是出现在操作员屏幕上的表格可见部分。然而, 它是你的程序可以使用的面向表格的信息, 要不然你就必须保持在你的程序中。为存取这种信息你可使用两种调用。(请见第三章关于命名的数据使用例子。)

1.10 终端键功能

许多表格驱动程序动作的采用是为了对操作员打入的某些终端键作出反应。FMS 已经对键赋予功能, 但你可以改变键和功能之间的对应关系以符合你的愿望。所以, 本手册中不涉及实际的键, 而只涉及键实现的功能(不管哪个键来实现此功能)。键功能的名字使用它的大写初始字母, 而不是它的整个名字, 后者是指出实际键名字的方法。(请见2.3节关于表格驱动程序使用的所有键的缺省定义。)

1.11 当 前 状 态

表格驱动程序建立了某些“当前状态”, 当你在书写你的程序时知道下列有关的情况是重要的:

1.当前终端——当前在用的终端(带所有它的特征)。

虽然在你的 FMS 应用中可以控制多个终端, 表格驱动程序调用每次只能影响一个终

端——当前终端。表格驱动程序提供改变当前终端的调用。

如果无终端在用，当前终端无定义。

2. 当前工作空间——当前在用的表格工作空间。

虽然在你的FMS应用中可以工作在多个工作空间上，但表格驱动程序调用每次只能影响一个工作空间——当前工作空间。你的程序可发调用来改变当前的工作空间。当前工作空间与当前终端相关联。每当你的程序转换到另一个终端，从而，它继承新终端的当前工作空间。

如果未定义当前终端，或者现时无工作空间与当前终端相关联时，当前工作空间无定义。

只有通过 LOAD、DISP或 CDISP 调用首先将表格装到工作空间，才能对它加工。

3. 当前字段——在供操作员输入的调用中指定的最当前字段（或者在尚未执行输入调用时，表格中的第一个可修改字段）和与此字段相关联的任何索引。

在随后的调用中你未为字段名变元提供名字时，按缺省通常使用当前字段。

当前字段和它的索引与当前的工作空间相关联。如果你的程序转换到另一个工作空间，它继承新的当前字段和索引。

如果字段没有索引属性，它的索引定义为零。

如果在表格中无字段或者未定义工作空间，当前字段和它的属性亦无定义。

4. 当前卷动行——表格驱动程序当前正在工作的任何卷动区域中的行。

此行最初是在卷动区域中的第一行（顶部行），但可在你的程序中发出 PFT 调用把此行在区域中向上或向下移动到一个新的当前卷动行。

当前卷动行是与当前工作空间中正在工作的任何卷动区域相关联的。如果在工作空间中的表格没有卷动区域，当前卷动行便无定义。

5. 最后结束符代码——在你的程序中通过调用返回的最新的字段结束符代码。

此代码与当前工作空间相关联。如果你的程序转换到另一工作空间，它继承新工作空间的最后结束符代码。

如果未曾为表格进入字段结束符，最后结束符代码便无定义。

6. 最后状态代码——由你程序中表格驱动程序调用返回的最新状态代码。

如果定义了这样的代码，它与当前工作空间相关联。如果未定义当前工作空间，则此代码与当前终端相关联（假定已定义了当前终端）。如未定义当前终端，此代码与表格驱动程序本身相关联。

把最后状态代码与多个可能的目标相关联的这种方法是有用的，以便在你的程序从一个终端转到另一个终端，或者从一个工作空间转到另一个工作空间时取得相应于它的上下文来说最新的状态代码，

7. 最后的 I/O 状态代码——由你的程序中表格驱动程序调用返回的 I/O 最新状态代码。

表格驱动程序用处理最后状态代码相同的方法来处理此代码。

8. 当前库通道——你程序需要存取相关联的表格库的任何时候要使用的 I/O 通道。

例如，你可以希望打开或关闭一个表格库，或者回收表格作处理。表格驱动程序使当前库通道。当前库通道与当前终端相关联。

9. 管理程序专用标志——控制具有“管理程序专用”属性的字段处理用的标志。

标志与当前终端相关联。如果当前终端无定义，则此标志也无定义。按缺省管理程序专

用标志为“置”状态。

10. 超时值——对从终端输入的调用，操作员必须作响应（用每个打入的字符）的时间（以秒计）。

如操作员在指定的时间里不能作出响应，此调用被夭折。此值与当前终端相关联，并初始化成零，表示无超时限起作用。如当前终端无定义，则超时值亦无定义。

11. 当前信号方式——表格驱动程序取得操作员关注的方式。

供你使用的有两种方式——终端响铃和逆转屏幕显示。逆转显示信号一直持续到操作员打入合法字符为止。当前信号方式与当前终端相关联。缺省方式是响铃（VT52 终端只有这种方式）。

1.12 操作员支持

1.12.1 帮助

当表格驱动程序等待输入时，操作员可以压帮助键请求帮助。可供使用的有帮助表格或者单帮助行。不管哪一种，当表格被设计时，提供的帮助也就被确定了。单行的帮助消息保持在屏幕的底部直到操作员压另一键为止。

如果帮助表格被显示，表格驱动程序等待操作员压 ENTER 键后才重新构造原始的屏幕。如果操作员不压 ENTER，而再次压帮助键，附加的帮助被显示（如果有附加帮助）。任何其它的输入不予理睬。（如果输入已被拒绝，则终端响铃或屏幕显示逆转。）

1.12.2 屏幕刷新

操作员通过压“刷新”键可更新屏幕图象。即，屏幕被清除，附加在终端并作显示的所有表格被重新显示。不再附加在终端，或标记为不作显示的所有表格（它们由以前的显示调用留在屏幕上）不重新显示。（你还可以从你的程序中发 RFRSH 调用。）

假如你的程序以前已发过 SPADA 调用（否则表格驱动程序不知道小键盘的状态）屏幕刷新还恢复小键盘方式。刷新操作还将终端的 LED 恢复到刷新发生之前它们所处的状态。

第二章 表格驱动程序的交互作用

表格驱动程序与你的程序以及与表格驱动程序相关联的终端交互作用。这种交互作用的讨论牵涉到你的程序在

- 内部加工表格
- 在终端上显示表格
- 征求终端操作员对一个接一个的表格数据请求的响应

等方面具有的控制程度。

在运行之前表格被设计的同时，对你的程序和操作员均作了某些限制。例如，对表格已经指定了字段属性，象在某些调用中字段被处理的顺序那样。

其它的考虑涉及到：

- 谁控制表格中字段的修改和什么时候控制
- 控制怎样传递给终端操作员
- 什么时候为操作员显示对字段和表格的不同级别的帮助
- 什么样的键和小键盘布局可供操作员在VT100/VT200和VT52等兼容终端使用。

2.1 与表格描述的交互作用

2.1.1 存贮和存取表格描述

你的程序可用以下两种方法之一存贮和存取表格描述：

- 作为磁盘留驻的表格，通过直接从已存贮在磁盘上的表格库文件读它们。
- 作为内存留驻的表格，将二进制的表格描述（作为目的模块存贮）与程序相联结。

两种方法均使用表格描述，它们是用表格编辑程序、表格语言翻译程序、表格升级实用程序或者表格转换程序建立的，并且已经过表格库管理程序或表格应用支持的处理。（请见《VAX FMS实用程序参考手册》。）

举例来说，当用表格编辑程序创建表格描述之后，必须使用FMS/LIBRARY/CREATE命令将描述存入表格库文件中，或者通过FMS/MEMORY_RESIDENT命令——一个表格应用支持——产生一个允许作为内存留驻表格来存取的目的模块。然后你必须将目的程序与表格驱动程序及任何其它你希望留驻内存的目的模块相联结，以形成可执行的FMS应用程序。（请见图2—1。）

注：

不一定要将一个应用中所有留驻内存的表格均构造在单个大目的模块中。通过FMS产生的多个留驻内存的表格目的模块可以联结在一个可执行映象中。