2000系列
软件资料

VAX TDMS
V1.4文件补充材料

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型存贮等特点，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系列机已在世界各地得到广泛应用，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、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23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2册，数据检索一卷6册，[illegible]卷85册。现已全部出版。

第二批资料包括VAX Rdb/VMS[illegible]七册）、VAX—11 LISP语言（三册）、VAX—11 CEP/VMS（五册）、VAX Cluster（一册）、VAX—11 RGL（四册）、Micro VAX（五册）、

本套第三批资料包括VAX [illegible]MS（[illegible]册）、VAX [illegible]DMS（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 TDMS V1.4文件补充材料**》（序号：AA—GS12A—TE）操作系统及版本：VMS MicroVMS，软件版本：VAX TDMS V1.5是由中国科学院沈阳计算所马哲翻译，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。

2000系列资料出版中心

一九八八年一月

目　　录

如何使用这本手册

新特点概述 ……2

RDU 性能增强 ……2

新的同步程序调用 ……3

异步程序调用 ……3

第一章　RDU性能增强

在核查状态下使用限定词/〔NO〕STORE ……4

在非核查状态下使用限定词/〔NO〕STORE……4

RDU 如何核查请求 ……4

BUILD LIBRARY/LIST/LOG命令 ……5

限定词/〔NO〕AUDIT ……5

第二章　TDMS 同步程序调用

TSS$COPY_SCREEN调用 ……6

TSS$DECL_AFK调用 ……8

TSS$UNDECT_AFK调用 ……11

TSS$WRITE_BRKTHRU调用 ……12

第三章　TDMS异步程序调用

TSS$CLOSE_A调用 ……16

TSS$COPY_SCREEN_A调用 ……18

TSS$DECL_AFK_A调用 ……21

TSS$OPEN_A调用……26

TSS$READ_MSG_LINE_A调用 ……28

TSS$REQUEST_A调用……31

TSS$UNDECL_AFK_A调用……35

TSS$WRITE_BRKTHRU_A调用 ……37

TSS$WRITE_MSG_LINE_A调用 ……39

如何使用这本手册

这本手册说明VAX—11终端数据管理系统（TDMS）1.4版本新的特点、限定词和程序调用，但仍称其为TDMS的手册。

VAX—11的数据检索简称为数据检索。

手册对象

这本手册面向想要了解TDMS1.4版本新内容的TDMS用户。用户应熟悉VAX TDMS 1.0到1.3版本的软件和文件集。

手册结构

这本参考手册包括三章：

第一章描述请求定义实用程序（RDU）性能的增强。

第二章包括关于TDMS1.4版本所不具备的同步程序调用的信息。

第三章包括关于TDMS异步程序调用的信息。

相关的手册

下述手册有助于阅读本书：

《VAX—11 TDMS概要描述》

《VAX—11 TDMS请求手册》

《VAX—11 TDMS格式手册》

《VAX—11 TDMS应用程序手册》

《VAX/VMS命令语言用户指南》或相应的VMS 4版本手册，VAX/VMS DCL词典

《VAX—11公共数据字典数据定义语言参考手册》

《VAX—11公共数据字典实用程序参考手册》

《VAX—11运行时间库参考手册》

为确保VMS操作系统的版本与本版本VAX—11 TDMS相匹配，请核查下述文献的最新版本：

- 对VMS操作系统——VAX/VMS选择软件交叉引用表，SPD25·99·XX
- 对Micro VMS操作系统——Micro VMX/Micro VMS选择软件交叉引用表，SPD 28·99·XX、

约定

本书采用下述约定：

大写	大写字母单词意味着该词不是变量
小写	小写字母单词指定是读者所提供的相应的词或词组。
[]	方括号中所包括的元素是可选的。
{}	花括号中所包括的元素中要求必选且只选1个。
{| |}	花括号内加竖杠所包括的元素中要求必选1个或更多的元素。
…	水平省略号意味着可以重复上一个元素。

表1列出第2,3章中所采用的TDMS参数传递标识。

表1. VAX运行时库参数传送标识

标　识	解　释
mz. r	可修改的未确定类型引用
rlu. r	只读的无符号长字逻辑类型，引用
rlu. v	只读的无符号长字逻辑类型，传值
rt. dx	只读的字符串，描述符类程类型
szem. r	无程序展开的调用，过程项掩码，引用
wlc. r	只写长字返回状态，引用
wlu. r	只写无符号长字逻辑类型，引用
wt. dx	只写字符串，描述符类程类型
wwu. r	只写无符号字逻辑类型，引用

表2列出TDMS返回及完成状态出错程度码，在第2,3章中将引用这些码。

表2.标志返回状态及结束状态的出错程度码：

出错程度	解　释
S	成功结束
I	只提供信息
F	指示一个严重错误；程序执行不续继，过程不产生任何输出结果

新特点概述

VAX TDMS 1.4版本包括下述新特点：

- 请求定义实用程序（RDU）性能增强；
- 新的同步程序调用；
- 异步程序调用。

RDU性能增强

TDMS允许你采用下述方式以大大节省建立请求库文件所需时间：

- 接受TDMS的缺省、核查方式及存贮方式。
- 当建立、修改、取代或核查一个请求，或核查请求库定义时，使用一个新的存贮限定词。

在上述任一种情况下，RDU都在公共数据字典（CDD）中存入相应的请求二进制位。只有当相关的格式或记录改变时，才采用一系列建造操作重新核查该请求。

在VALIDATE REQUEST 和VALIDATE LIBRARY命令中增加/〔NO〕AUDIT限定词。/〔NO〕AUDIT限定词在这里的作用与对其它RDU命令的作用相同。

新的同步程序调用

TDMS 1.4版本包括下述新的同步程序调用：

- TSS$UNDECL_AFK
- TSS$WRITE_BRKTHRU

上述调用遵循TDMS 1.0—1.3版本中程序调用的同样规则。

异步程序调用

TDMS 1.4版本包括下述异步程序调用：

- TSS$CLOSE_A
- TSS$COPY_SCREEN_A
- TSS$DECL_AFK_A
- TSS$OPEN_A
- TSS$READ_MSG_LINE_A
- TSS$REQUEST_A
- TSS$UNDECL_AFK_A
- TSS$WRITE_BRKTHRU_A
- TSS$WRITE_MSG_LINE_A

上述程序调用是为有经验的程序员进行复杂的应用设计而提供的。当你进行异步调用时，该调用被初始化，并立刻将控制返回应用程序。

对于异步程序调用，TDMS返回的标准VAX条件码只说明该调用被初始化，而不说明该调用完成。你可以在你的应用程序中用蕴含代码（including code）来测试返回状态。异步程序调用还返回特殊的完成码（rsb）。在“TDMS异步程序调用”这一章中再讨论这种完成码及其它异步参数。

第一章 RDU 性能增强

TDMS1.4版本使你大大节省用于建立请求程序库文件的时间。在核查（TDMS缺省）状态下，当你创建、修改或替换一个请求时，RDU仍要核查该请求并把该请求定义留给CDD。只有当相关的表格或记录被改变时，才需由一系列建选操作重新核查该请求。

对请求标志存贮的缺省值依核查状态而定。在下述情况下RDU处于核查状态。

- 接受TDMS缺省值
- 使用下列任一条命令
  —SET VALIDATE
  —VALIDATE LIBRARY
  —VALIDATE REQUEST

还可以在下述任一条命令中使用限定词/〔NO〕STORE，从而使RDU置于存贮/非存贮状态：

- CREATE REQUEST
- MODIFY REQUEST
- REPLACE REQUEST
- VALIDATE REQUEST
- VALIDATE LIBRARY

在核查状态下使用限定词/〔NO〕STORE

在核查状态下，如果在CREATE REQUEST、MODIFY REQUEST、REPLACE REQUEST或VALIDATE REQUEST中使用限定词/STORE，RDU就将相应的请求二进位存入CDD。如果在VALIDATE LIBRARY命令中使用限定词/STORE，RDU就为每一个包含在请求库定义中的请求存入请求二进位。

如果当RDU处于核查状态下时使用限定符/NOSTORE，RDU不在CDD中存贮任何请求二进位。这时CDD的尺寸减小，但建立请求库文件的时间增加。

核查状态的缺省值是/STORE。

在非核查状态下使用/〔NO〕STORE

如果在非核查状态下创建、修改或取代一个请求，RDU不对该请求核查，也不在CDD中存贮相应的请求二进位。例如：如果在一个请求中的格式或记录尚未创建，就应采用非核查状态。

RDU如何核查请求

请求核查过程包括三个阶段：

1. 如果该请求以前核查过，RDU则检查相应的格式，记录是否改变了。如果未改变，该请求是合法的。
2. 如果该请求以前未核查过，或相应的格式，记录改变了，RDU就核查该请求。
3. RDU将请求标志存入CDD。只有在下述情况下这一动作才执行：
   - 必须进行第2阶段工作
   - 核查处理完全成功

· RDU处于存贮状态。

BUILD LIBRARY/LIST/LOG命令

如果请求二进位已存入CDD并且仍有效，RDU就不列出%ALL映象。如果请求二进位未存在CDD中，或者请求二进位存在CDD中但必须重新核查（例如在上次核查后相关的格式或记录又改变了），RDU就列出%ALL映象。

注意：这个限制只适用于BUILD LIBRARY/LIST/LOG命令。

限定词/〔NO〕AUDIT

限定词/〔NO〕AUDIT可附属于命令VALIDATE RERUEST和VALIDATE LIBRARY，因为这二个命令可以借助在CDD中加请求二进位来修改请求或请求库定义。限定词/〔NO〕AUDIT在此所起的作用与在其它RDU命令中的作用相同。

限定符/〔NO〕AUDIT的缺省值是/AUDIT。

第二章　TDMS 同步程序调用

本章对全部TDMS 1.4版的新的同步程序调用提供方便的索引。这些调用按字母表顺序排列。对所有调用说明如下：

格 式　调用的一般格式采用VAX—11运行时库参数传递标记。关于这些标记参见表1。

调用参数　描述传送变元

返回状态　关于返回状态以及返回码表的一般描述。另外还可见关于返回状态的出错程度码。关于出错程度码及其所指的结果参见表2。

注释　如何使用该调度的附加信息。

举例　用BASIC、COBOL和FORTRAN

本章结尾处列出一个用语言概述全部TDMS新的同步调用表。

TSS$COPY_SCREEN调用

将当前屏幕内容复制到指定文件或由逻辑TSS$HARDCOPY定义的文件。

格式

```
ret—status.wlc.v=TSS$COPY_SCREEN(channel.rlu.r
                                 [,file—spec.rt.dx]
                                 [,append—flag.rlu.r])
```

调用参数

channel

包含TDMS通道号的长字地址，该通道号是唯一的，由TSS$OPEN调用分配的。通过引用来传递该参数。

file—spec

指向VMS文件说明的串描述符地址，该文件说明由屏幕内容指定。通过描述符传送这个可任选的参数。

如果该参数未指定，则用逻辑的TSS$HARDCOPY值决定屏幕内容在哪里。

如果该参数未指定且逻辑TSS$HARDCOPY未定义，该调用不做任何事。返回信息状态：TSS$_NOOUTFILE。

append—flag

一个长字地址，该长字决定是创建该文件的一个新版本还是将屏幕内容拼接在文件最新版本之后。通过引用传送该任选的参数。如果选用该参数，其值必须为0或1。

在下述情况下将创建输出文件的新版本：

- 参数不存在
- flag值为0
- 尚无该文件版本

否则，当前屏幕内容将接续在现存VMS文件最新版本之后。

返回状态

ret—status 是VAX/VMS的标准返回状态，它指出本次调用的成败。调用后返回的代码是：

TSS$_BADFLAGS
标志参数值无效（F）
TSS$_BUGCHECK
内部软件致命错误（F）
TSS$_CANCEL（F）
用TSS$_CANCEL（F）撤消调用
TSS$_CANINPROG
在通道上撤消对该调用的处理（F）
TSS$_COPYOUTERR
TSS$COPY_SCREEN输出错（F）
TSS$_INSVIRMEM
虚存空间不够（F）
TSS$_INVARG
非法变元（F）
TSS$_INVCHN
非法通道（F）
TSS$_INVDSC
非法描述符（F）
TSS$_IOINPROG
在通道上进行I/O（F）
TSS$_NOOUTFILE
输出文件无定义（I）
TSS$_NORMAL
正常完成（S）
TSS$_SYNASTLVL
在AST级上不能调用同步调用（F）

注释

TSS$COPY_SCREEN给程序提供了可调用的PF4函数。当在某个具体通道上进行输出时，或TSS$REQUEST以及TSS$WRITE_MSG_LINE调用尚未完成时，不能发TSS$COPY_SCREEN调用。

提示：同步与异步调用可以混用。例如：TSS$OPEN_A可与TSS$CLOSE同用，TSS$DECL_AFK_A可与TSS$UNDECL_AFK同用。

举例

BASIC

```
RET_STATUS=TSS$COPY_SCREEN( CHANNEL, &
                            FILE_SPEC, &
                            APPEND_FLAG )
```

COBOL

```
CALL"TSS$COPY_SCREEN"
     USING BY REFERENCE CHANNEL,
           BY DESCRIPTOR FILE-SPEC,
           BY REFERENCE APPEND-FLAG,
     GIVING STATUS-RESULT.
```

FORTRAN

```
RET_STATUS=TSS$COPY_SCEREN(%REF(CHANNEL),
1                          %DESCR("FILE_SPEC"),
2                          %REF(APPEND_FLAG))
```

TSS$DECL_AFK调用

这种调用指定实用功能键（AFK）并将它们与服务子程序或事件标记相关连，从而使终端操作员可以利用这些键。

格式

```
ret-status.wlc.v=TSS$DECL_AFK(channel.rlu.r /
                              key-id.rlu.r
                               {| /key-efn.rlu.r      |}
                               {|                     [}
                               {| /key-astadr.szem.r  |}
                               {|{ /key-astprm.rlu.v } |}
```

调用参数

channel

包含TDMS通道号的长字地址，该通道号是唯一的，由TSS$OPEN调用分配的，通过引用来传递该参数。

key—id

包含代表AFK代码的长字地址，可通过引用来传递该参数。当操作员按下代表key—id的功能键时，应用程序将会被通知。关于实用功能键参见表2—1。

因为key—id是个代码，所以每次调用TSS$DECT_AFK只能处理一个键。

key—efn

包含事件标记号的长字地址，该标记号是当终端用户按AFK时置入的。该参数通过引用传送。若参数不出现，当终端操作员按功能键时TDMS不置事件标记。

对事件标记，可以单独使用，也可与AST服务子程序结合使用。

key—astadr

应用程序中子程序的地址，通过引用传递该参数。当终端操作员按下上述AFK时，VAX TDMS在AST级调用该子程序。用户子程序必须具备下述调用序列：

```
status  wlc. v=ROUTADR(key-astprm. rlu. v
                       /channel. rlu. r
                       /key-id. rlu. r)
```

使用AST服务子程序用不用AST参数均可。使用AST服务子程序不可带事件标记。

key—astprm

包含AST参数的长字，该参数要传送给AFK服务子程序。该参数可选，通过值传送。如AST参数未出现，而相应的服务子程序存在，TDMS就传送0值参数给服务子程序。

表2—1 TDMS实用功能键（AFKs）

Key Id	控 制 键	key Id	控 制 键
0	CTRL/space bar	15	CTRL/O
1	CTRL/A	16	CTRL/P
2	CTRL/B	18	CTRL/R
3	CTRL/C	20	CTRL/T
4	CTRL/D	21	CTRL/U
5	CTRL/E	22	CTRL/V
6	CTRL/F	23	CTRL/W
7	CTRL/G	24	CTRL/X
8	CTRL/H	25	CTRL/Y
9	CTRL/I	26	CTRL/Z
10	CTRL/J	27	CTRL/[
11	CTRL/K	28	CTRL/backslash
12	CTRL/L	29	CTRL/]
13	CTRL/M	30	CTRL/tilde
14	CTRL/N	31	CTRL/?

TDMS将该参数作为值处理： 你愿意自己的AST子程序所接受的任意类型参数均可传送，包括地址（通过引用传递的参数）。

返回状态

ret—status是指出调用成败的标准VAX/VMS返回状态。可返回的代码如下：

TSS$_BUGCHECK

内部软件致命错（F）

TSS$_INSVIRMEM

虚存空间不足（F）

TSS$_INVARG

非法变元（F）

TSS$_INVCHN

非法通道（F）

TSS$_INVKEYID

非法key id（F）

TSS$_NORMAL

正常结束（S）

TSS$_SYNASTLVL

在AST级上不可进行同步调用（F）

注释

实用功能键（AFK）提供处理终端相关事件的例外通知服务。在执行TDMS应用期间，终端操作员按下AFK键，从而使当前输入环境之外的动作初启为活跃格式。

AFK是异步功能键，即它们互相独立地处理请求。作为异步功能键，AFK在用户应用程序中初启异步处理过程。

应用程序员在应用程序中通过发TSS$DECT_AFK和TSS$UNDECL_AFK调用来授权和禁止终端操作员使用AFK。TSS$DECL_AFK通过key_id指定AFK并将该键与服务子程序或事件标记相关连。程序执行TSS$DECL_AFK调用后，终端操作员若要起用一个特殊功能，就可随时按下被授权使用的键，直到该键在下述情况下被禁用：

- 应用程序发相应的TSS$UNDECL_AFK或TSS$UNDECL_AFK_A调用；
- 应用程序用TSS$CLOSE或TSS$CLOSE_A调用关闭终端；
- 应用程序结束。

当按下AFK，TDMS做如下动作之一：

- 若有事件标记则将其指定给该AFK
- 若有服务子程序则在AST级调用它，并将下述参数传递给它：
  - AST参数
  - 按AFK的那个通道
  - 所按AFK的key—id
- 既置事件标记又调服务子程序。在这种情况下TDMS先置事件标记后调服务子程序。

无论是否还有尚未处理完的请求，一旦VMS终端驱动器处理AFK键的键入，TDMS就调用服务子程序。

除CTRL/Q和CTRL/S外，所有控制键均可定义为AFK。

注意： 某些控制键（如CTRL/H或BACK SPACE，CTRL/I或TAB）还可以在请求中定义为程序请求键（PRK）。如果同一键既定义为AFK又定义为PRK，则用AFK处理取代PRK处理。

TDMS应用的设计者在为AFK或PRK设计多重功能时应当心，否则可能产生意外结果。

注意： 同步与异步调用可混用。例如：TSS$OPEN_A与TSS$CLOSE可共同，TSS$DECL_AFK_A与TSS$UNDECL_AFK可共用。

举例

BASIC

```
EXTERNAL LONG KEY_AST_ROUTINE

RET_STATUS=TSS$DECL_AFK(CHANNEL ,&
                        KEY_ID ,&
                        KEY_EVENT_FLAGNUMBER ,&
                        LOC(KEY_AST_ROUTINE) BY VALUE ,&
                        KEY_AST_PARAMETER BY VALUE)
```

COBOL

```
CALL"TSS$DECL_AFK"
```

```
    USING BY REFERENCE CHANNEL ,
          BY REFERENCE KEY_ID ,
          BY REFERENCE KEY-EVENT-FLAG-NUMBER ,
          BY REFERENCE KEY-AST-ROUTINE ,
          BY VALUE KEY-AST-PARAMETER.
    GIVING STATUS-RESULT.
```

FORTRAN

```
 RET_STATUS=TSS$DECL_AFK ( %REF ( CHANNEL ) ,
1                          %REF ( KEY_ID ) ,
2                          %REF ( KEY_EVENT_FLAG_NUMBER ) ,
3                          %REF ( KEY_AST_ROUTINE ) ,
4                          KEY_AST_PARAMETER )
```

TSS$UNDECL_AFK调用

禁用实用功能键（AFK）和相关的服务子程序及事件标记。

格式

```
ret_status. wlc. v=TSS$UNDECL_AFK ( channel .rlu. r,
                                    key-id. rlu. r )
```

调用参数

channel

包含通道号的长字地址，该通道号是唯一的，由TSS$OPEN 调用分配的。通过引用传递该参数。

key-id

包含代表AFK代码的长字地址，该AFK应用程序不再需要。通过引用传递该参数。

返回状态

ret-status 是指示调用成败的标准VAX/VMS返回状态。可返回的代码是：

TSS$_BUGCHECK
内部软件致命错（F）

TSS$_INSVIRMEM
虚存空间不足（F）

TSS$_INVARG
非法变元（F）

TSS$_INVCHN
非法通道（F）

TSS$_INVKEYID
非法key id（F）

TSS$_NORMAL
正常结束（S）

TSS$_SYNASTLVL
在AST级不可进行同步调用（F）

注释

这个调用用来挂起AFK的异步通知，TSS$UNDECL_AFK调用发出后，TDMS将不再调用由相应TSS$DECL_AFK调用所指定的服务子程序。即在TSS$UNDECL_AFK调用后，当终端操作员按该功能键时也不再通知程序。

注意：同步与异步调用可以混用。如TSS$OPEN_A与TSS$CLOSE可以共同，TSS$DECL_AFK_A与TSS$UNDECL_AFK可以共同。

举例

BASIC

```
RET_STATUS=TSS$UNDECL_AKF(CHANNEL, &KEY_ID)
```

COBOL

```
CALL "TSS$UNDECL_AFK"
     USING BY REFERENCE CHANNEL,
           BY REFERENCE KEY-ID,
     GIVING STATUS-RESULT.
```

FORTRAN

```
 RET_STATUS=TSS$UNDECL_AFK(%REF(CHANNEL)
1                          %REF(KEY_ID))
```

TSS$WRITE—BRKTHRU调用

在终端的保留消息行上写消息。为此，将当前请求或消息行操作中断。

格式

```
ret-statns. wlc. v=TSS$WRITE_BRKTHRU(channel. rlu. r
                                     . message-text. rt. dx
                                     . (bell-flag. rlu. r))
```

调用参数

channel

包含TDMS通道号的长字地址，该通道号是唯一的，由TSS$OPEN调用分配的。通过引用传送该参数。

message—text

串描述符地址，该描述符指向将在消息行显示的正文。该参数由描述符传递。

bell—flag

包含终端铃标记的长字地址。该参数通过引用传递。若置1，当消息正文显示时该标记使终端响铃。如果不传递该参数或该参数值为0，TDMS不使铃响。

返回状态

ret—status是指示调用成败的标准VAX/VMS返回状态。能返回的代码是：

TSS$_BADFLAGS

标记参数值非法（F）

TSS$_BUGCHECK

内部软件致命错（F）

TSS$_INSVIRMEM

虚存空间不足（F）

TSS$_INVARG

非法变元（F）

TSS$_INVCHN

非法通道（F）

TSS$_INVDSC

非法描述符（F）

TSS$_NONPRICHA

字段中包含了非打印字符（F）

TSS$ _NORMAL

正常结束（S）

TSS$_SYNASTLVL

在AST级不可进行同步调用（F）

注释

如果本调用的消息正文超出当前终端行尺寸则被截断。

注意：同步与异步调用可混用。如TSS$OPEN_A和TSS$CLOSE可共用，TSS$DECL_AFK_A和TSS$UNDECL_AFK可共用。

举例

BASIC

```
RET_STATUS=TSS$WRITE_BRKTHRU(CHANNEL,&
                             MESSAGE_TEXT,&
                             BELL_FLAG)
```

COBOL

```
CALL"TSS$WRITE_BRKTHRU"
    USING BY REFERENCE CHANNEL,
          BY DESCRIPTOR MESSAGE-TEXT,
          BY REFERENCE BELL_FLAG,
    GIVING STATUS-RESULT.
```

FORTRAN

```
RET_STATUS=TSS$WRITE_BRKTHRU(%REF(CHANNEL),
1                            %DESCR("MESSAGE_TEXT"),
2                            %REF(BELL_FLAG))
```

表2—2、2—3、2—4列出BASIC、COBOL、FORTRAN语言形式的新的TDMS同步程序调用。

表2—2 用VAX—11 BASIC表示的新的TDMS同步程序调用

调　　用	格　　式
TSS$COPY_SCREEN	RET_STATUS=TSS$COPY_SCREEN& (CHANNEL,& FILE_SPEC,& APPEND_FLAG)
TSS$DECL_AFK★	RET_STATUS=TSS$DECL_AFK& (CHANNEL,& KEY_ID,& KEY_EVENT_FLAG_NUMBER,& LOC(KEY_AST_ROUTINE)BYVALUE,& KEY_AST_PARAMETER BY VALUE)

续上表

调　　用	格　　式
TSS$UNDECL_AFK	RET_STATUS=TSS$UNDECL_AFK& (CHANNEL,& KEY_ID)
TSS$WRITE_BRKTHRU	RET_STATUS=TSS$WRITE_BRKTHRU& (CHANNEL,& MESSAGE_TEXT, & BELL_FLAG

表2—2注释：

★在BASIC程序开头需要下述说明：

EXTERNAL LONG KEY_AST_ROUTINE

表2—3　用VAX—11 COBOL表示的新的TDMS同步程序调用

调　　用	格　　式
TSSSCOPY_SCREEN	CALL"TSS$COPY_SCREEN"USING BY REFERENCE CHANNEL, BY DESCRIPTOR FILE_SPEC, BY REFERENCE APPEND—FLAG GIVNIG STATUS_RESULT.
TSS$DECL_AFK	CALL"TSS$DECL_AFK"USING BY REFERENCE CHANNEL BY REFERENCE KEY—ID, BY REFERENCE KEY—EVENT—FLAG—NUMBER, BY REFERENCE KEY—AST—ROUTINE. BY VALUE KEY_AST—PARAMETER, GIVING STATUS—RESULT.
TSS$UNDECL_AFK	CALL"TSS$UNDECL_AFK"USING BY REFERENCE CHANNEL, BY REFERENCE KEY—ID, CIVING STATUS—RESULT.
TSS$WRITE_BRKTHRU	CALL"TSS$WRITE_BRKTHRU"USING BY REFERENCE CHANNEL, BY DESCRIPTOR MESSAGE—TEYT, BY REFERENCE BELL—FLAG, GIVING STATUS—RESULT.

表2—4 用VAX—11 FORTRAN表示的新的程序TDMS同步程序调用

调　　用	格　　式
TSS$COPY_SCREEN	RET_STATUS=TSS$COPY_SCREEN 1 (%REF(CHANNEL), 2 %DESCR("FILE_SPEC"), 3 %REF(APPEND_FLAG))
TSS$DECL_AFK	RET_STATUS=TSS$DECL_AFK 1 (%REF(CHANNEL), 2 %REF(KEY_ID), 4 %REF(KEY_EVENT_FLAG_NUMBER), 5 %REF(KEY_AST_ROUTINE), 6 KEY_AST_PARAMETER)