

VFP表跟踪日志的实现方法 PDF转换可能丢失图片或格式，  
建议阅读原文

[https://www.100test.com/kao\\_ti2020/273/2021\\_2022\\_VFP\\_E8\\_A1\\_A8\\_E8\\_B7\\_9F\\_E8\\_c97\\_273097.htm](https://www.100test.com/kao_ti2020/273/2021_2022_VFP_E8_A1_A8_E8_B7_9F_E8_c97_273097.htm) 本文介绍了Visual Foxpro中数据库触发技术的应用实例，为VFP表建立跟踪日志,提供解决网络多用户环境中数据文件安全的一种方法。关键词 触发技术，表跟踪，自定义函数 表跟踪问题的提出：在管理信息系统的正常运行中，信息的安全性问题日益突出。虽然网络为我们提供了一定的数据安全性，只有授权的用户才能对数据进行日常的维护，但有时用户的误操作可能导致数据的丢失或混乱；另外，有时我们需要对数据的改动留下记载等。对重要的数据文件，记录每一个数据变化的审核日志，对其建立其相应的日志表来跟踪对它的增、删和修改操作是十分必要的。日志表中记载了何时、何用户进行了何种操作，记载了被用户增加或删除的记录信息，以及被修改记录的原来信息和修改后的信息，将被操纵过的记录进行备份。VFP触发技术：Visual Foxpro与它的前期版本重大差别之一是使用数据库（Database）这一强大功能，使之成为一个真正的DataBase。VFP的Database Container包含了表（Table）及表的各种属性定义，也包含了属于后端的SQL中的触发技术（Trigger）。Trigger的使用为Database中的Table提供了一个记录级规则（即检测程序）。Trigger分为Insert Trigger、Update Trigger和Delete Trigger三种操作。Insert Trigger用于Table新增记录时所触发的检测程序，Update Trigger用于Table修改记录回存时所触发的检测程序，Delete Trigger用于Table中记录被删除时所触发的检测程序。VFP允许对任一种操作设定不同的

规则，该规则可以是表达式或用户自定义函数（UFD），若响应为真值时，表示触发成功，若响应为假值时，表示触发失败。用户自定义函数（UFD）存储在Database Container中。创建触发器有两种方式：1) 打开表属性”对话框，在INSERT、UPDATE或DELETE触发器框中，输入触发器表达式或用户自定义函数（UFD）；2) 使用CREATE TRIGGER命令。创建、修改或移去存储过程有三种方式：1) 在项目管理器中，选择并展开一个数据库，选定“存储过程”，然后选择“新建”、“修改”或“移去”按钮；2) 在数据库设计器中，从“数据库”菜单中选择“编辑存储过程”按钮；3) 在COMMAND命令窗口中，使用MODIFY PROCEDURE命令。

表跟踪设计：利用Insert Trigger、Update Trigger和Delete Trigger三种操作原理，我们可以构造备，以实现表跟踪。这样，不论用户使用程序进行对此表的操纵或直接打开此表进行增删和修改记录，都能激发Trigger将被操纵记录进行日志记载。本人构造了四个通用的自定义函数，Inslog()、Dellog()、Modlog()和addlog()，存放于Database Container的存储过程中。在需要进行表跟踪的数据文件的Insert Trigger、Delete Trigger和Update Trigger中分别加入Inslog()函数、Dellog()函数和Modlog()函数，以使对数据文件中记录的增、删和修改均激发Trigger进行日志记载。addlog()用于建立被跟踪表的日志表结构，日志表的结构设计包含了被跟踪表的所有字段信息，并且增加了用户标志opid、操作日期时间optime和操作标志opmak字段。日志表的命名规则为被跟踪表的表名前加‘L,以示区分，另外本人在存放数据文件目录下另设子目录LOGDBFS，专用于存放日志表以便管理。这四

```

个函数程序设计如下：*Insert Trigger:FUNCTION inslog()
m.path=substr(dbf(),1,at(\,dbf()))m. logname=m.path
LOGDBFS\L SUBSTR(DBF(),LEN(m.path) 1)IF NOT
FILE(m.logname)=addlog() amp.建立日志表结
构ENDSCATTER MEMVARm.opmak=增 amp.增加记录标
志m.opid=SYS(0) amp.取网络用户标志m.optime=datetime()
amp.增加记录时间INSERT INTO (m.logname) .FROM
MEMVAR amp.记载所增记录RETURN .T.*Delete
Trigger:FUNCTION dellog() m.path=substr(dbf(),1,at(\,dbf()))m.
logname=m.path LOGDBFS\L SUBSTR(DBF(),LEN(m.path) 1)IF
NOT FILE(m.logname)=addlog() ENDSCATTER
MEMVARm.opmak=删 amp.删除记录标
志m.opid=SYS(0)m.optime=datetime()INSERT INTO
(m.logname) .FROM MEMVAR amp.记载所删记录RETURN
.T.*Update Trigger:FUNCTION modlog()
m.path=substr(dbf(),1,at(\,dbf()))m. logname=m.path
LOGDBFS\L SUBSTR(DBF(),LEN(m.path) 1)IF NOT
FILE(m.logname)=addlog()ENDIDIME alog[FCOUNT()
3]alog[FCOUNT() 2]=SYS(0)alog[FCOUNT()
3]=datetime()alog[FCOUNT() 1]=修(原) FOR i=1 TO
FCOUNT()alog[I]=oldval(field[I])ENDFORINSERT INTO
(m.logname) .FROM ARRAY alog amp.记载更改前信
息alog[FCOUNT() 1]=修(现) FOR i=1 TO
FCOUNT()alog[I]=EVALUATE(field[I])ENDFORINSERT
INTO (m.logname) .FROM ARRAY alog amp.记载更改后信
息RETURN .T.*建立日志表结构:FUNCTION addlog()*取被跟

```

踪表字段COPY STRU TO tmp EXTEND \*建立用户标志、操作时间和操作标志字段信息INSERT INTO tmp.dbf  
. (field\_name,field\_type,field\_len).VALUES (OPMAK,C,6)INSERT INTO tmp.dbf .(field\_name,field\_type,field\_len).VALUES (OPID,C,10)INSERT INTO tmp.dbf .(field\_name,field\_type,field\_len).VALUES (OPTIME,T,8)USE IN tmpold0select=SELECT() amp.生成日志表结构SELE 0CREATE (m.logname) FROM tmp DELETE FILE tmp.dbfSELECT (old0select) 100Test 下载频道开通 , 各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)