VA FILEMAN

TECHNICAL MANUAL

Version 22.0

March 1999

Department of Veterans Affairs
VHA OI Health System Design & Development (HSD&D)
Infrastructure and Security Services (ISS)

Preface

The purpose of the Technical Manual is to provide information about the technical structure of VA FileMan. It also contains material specifically intended for VA's Veterans Health Information Systems and Technology Architecture (**V***IST***A**) systems managers, application programmers, and developers. It contains information about the routines and files that comprise VA FileMan. It also has information about VA FileMan's structure and recommendations regarding VA FileMan's efficient use (e.g., routine mapping). However, it does *not* describe how VA FileMan is used nor does it detail its use in package development.

The *VA FileMan Technical Manual* is part of a multi-manual set that also contains the following manuals:

- *VA FileMan V. 22.0 Release Notes* (PDF format)
- *VA FileMan V. 22.0 Installation Guide* (PDF format)

  This guide also includes instructions for installing a stand-alone implementation.

- *VA FileMan V. 22.0 Getting Started Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5

- *VA FileMan V. 22.0 Advanced User Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5

- *VA FileMan V. 22.0 Programmer Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5

  This manual describes how VA FileMan is used and details its use in package development.

Together, these manuals document the features, use, and structure of VA FileMan.

Table of Contents

Orientation

Installation of VA FileMan in the Veterans Health Information Systems and Technology Architecture (**V**_IST_**A**) environment is described in the *VA FileMan Installation Guide*.

How to Use this Manual

This manual uses several methods to highlight different aspects of the material:

- Descriptive text is presented in a proportional font (as this is).
- "Snapshots" of computer online displays (i.e., roll-and-scroll screen captures/dialogues) and computer source code are shown in a *non*-proportional font and enclosed within a box.
  - User's responses to online prompts will be boldface.
  - The "**<RET>**" found within these snapshots indicate that the user should press the Enter or Return key on their keyboard.
  - Author's comments are displayed in italics or as "callout" boxes (callout boxes refer to labels or descriptions, usually enclosed within a box, which point to specific areas of a displayed image).
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).
- Two symbols are used throughout the documentation to alert the reader to special information. The following table gives a description of each of these symbols:

Symbol	Description
ℹ	Used to inform the reader of general information including references to additional reading material.
⚠	Used to caution the reader to take special notice of critical information.

Table 1: Documentation Symbol Descriptions

How to Obtain Technical Information Online

Help at Prompts

To retrieve online documentation in the form of Help in VA FileMan, entering a single question mark ("**?**") at a field/prompt shows the contents of two data dictionary fields:

- HELP PROMPT—A FREE TEXT field for a brief description.
- EXECUTABLE HELP—Provides additional Help text, if defined. This field can contain M code.

Entering two question marks ("**??**") at a field/prompt also shows the contents of two data dictionary fields:

- DESCRIPTION—A FREE TEXT field for a more detailed description.
- EXECUTABLE HELP—Provides additional Help text, if defined. This field can contain M code.

If a field is a pointer, entering one question mark ("**?**") displays the HELP PROMPT and a list of choices, if the list is short. If the list is long, the user will be asked if the entire list should be displayed. A YES response will invoke the display. The display can be given a starting point by prefacing the starting point with an up-arrow ("**^**") as a response. For example, **^M** would start an alphabetic listing at the letter M instead of the letter A while **^127** would start any listing at the 127th entry. Two question marks (**??**) will display the entire list without the "DO YOU WANT TO SEE THE ENTIRE LIST?" prompt.

TECHNICAL DESCRIPTION is another data dictionary attribute that provides technical information. It can only be displayed by using the List File Attributes option. The TECHNICAL DESCRIPTION is not shown when entering question marks at the field level.

For more information on the List File Attributes option, please refer to the "List File Attributes" chapter in the "File Management" section of the *VA FileMan Advanced User Manual.*

Obtaining Data Dictionary Listings

Information about files and the fields in files is stored in data dictionaries. You can use the List File Attributes option on the Data Dictionary Utilities submenu to print formatted data dictionaries.

For details about obtaining data dictionaries and about the formats available, please refer to the "List File Attributes" chapter in the "File Management" section of the *VA FileMan Advanced User Manual*.

Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the **V***IST***A** computing environment.

No attempt is made to explain how the overall **V***IST***A** programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web (WWW) for a general orientation to **V***IST***A**. For example, check out the following web sites:

- **V***IST***A** Software Development Home Page: http://vaww.vista.med.va.gov/

Related Manuals and Other References

Readers who wish to learn more about VA FileMan should consult the following:

- *VA FileMan V. 22.0 Release Notes* (PDF format)
- *VA FileMan V. 22.0 Installation Guide* (PDF format)
- *VA FileMan V. 22.0 Getting Started Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5
- *VA FileMan V. 22.0 Advanced User Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5
- *VA FileMan V. 22.0 Programmer Manual* (HTML format)

  http://www.va.gov/vdl/Infrastructure.asp?appID=5

Readers who wish to learn more about VA FileMan should consult the VA FileMan Home Page at the following web address:

http://vista.med.va.gov/fileman/index.asp

ℹ This site contains additional information and documentation.

VA FileMan documentation is made available online (**HTML** format), on paper, and in Adobe Acrobat Portable Document Format (**.PDF**).

The **HTML** documents must be read using a web browser (e.g., Microsoft Explorer or Netscape Navigator, both of which are freely distributed). Using the web browser, open the "table of contents" page (i.e., INDEX.HTML). The distinguishing characteristic of manuals in this format are the *hypertext jumps* contained within the text. Clicking on a hypertext jump causes your browser to jump to the location or document described in the jump. The *VA FileMan Getting Started Manual*, the *VA FileMan Advanced User Manual*, and the *VA FileMan Programmer Manual* are all linked together.

The **.PDF** documents must be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is also freely distributed by Adobe Systems Incorporated at the following web address:

http://www.adobe.com/

Introduction

VA FileMan is a database management system (DBMS) consisting of computer routines written in American National Standards Institute (ANSI) Standard M, along with associated files. Developed with portability as a goal, VA FileMan runs on all major implementations of ANSI M and on hardware platforms ranging from PCs to mainframes.

Programmers and non-programmers use VA FileMan alike. VA FileMan can be used as a stand-alone database or as a set of application utilities. In either mode, it is used to define, enter, and retrieve information from a set of computer-stored files, each of which is described by the data dictionary.

VA FileMan is a public domain software package and is widely used in clinical, administrative, and business settings in the United States and abroad.

Implementation and Maintenance

VA FileMan is initialized with the DINIT routine, followed by an install using the Kernel Distribution and Installation system (KIDS) as directed in the *VA FileMan Installation Guide*. DINIT only needs to be run once; it is nondestructive to a system if run again. Stand-alone VA FileMan sites do not perform the KIDS install.

VA FileMan routines and globals occupy approximately 3.5 MB of memory. The size of the globals, particularly those that store file data, will increase when VA FileMan is used.

Since VA FileMan provides the DBMS upon which all files in **V***IST***A** are based, it must be present on all **V***IST***A** systems. The current version of VA FileMan is designed for complete backward compatibility; files and applications developed under prior versions will remain usable.

If used with Kernel, all or part of the VA FileMan options can be given to users. Those who are able to use programmer mode can also invoke the main menu from the M prompt. Of course, anyone can use applications developed with VA FileMan, whether or not direct access to VA FileMan itself is allowed.

For more information on programmer mode, please refer to the "^DI: Programmer Access" chapter in the Developer's Tools section of the *VA FileMan Programmer Manual*.

When used with Kernel, VA FileMan allows the user to print multiple copies. In order to do this, a temporary storage location must be allocated on the system with a corresponding DEVICE file entry that uses a sequential disk processor (SDP) device type.

The *Kernel Systems Manual* contains specific instructions on how to set up an SDP device for different operating systems.

The ^DISV global contains the most recent lookup value for files and subfiles; it is used to process <Spacebar Return> input. The ^DOSV global contains results of statistical operations. With time, these globals can grow to considerable size. They should be monitored. It is safe to periodically kill these globals. Users should not be logged on to the system when the globals are killed in order to minimize inconvenience and avoid data corruption.

The site manager must monitor the proliferation of routines with names like ^DISZnnnn where "nnnn" is a four-digit number with leading zeros. These routines are created when compiled sorts are run. Ordinarily, they are deleted after the sort completes, but, if the system goes down or the job fails with an error, they may remain. When users are not on the system, the routine ENRLS^DIOZ can be run to clean up these routines and to release the "nnnn" numbers for reuse.

For more information on the ENRLS^DIOZ utility, please refer to the "COMPILED ROUTINE File Cleanup: ENRLS^DIOZ( )" topic in the "System Management" chapter in the "Tools" section of the *VA FileMan Advanced User Manual*.

Routine Descriptions and Callable Routines (Supported Entry Points)

The routines that comprise VA FileMan are listed in this section. Brief descriptions are given in the "Routine Description" column below. Routines of VA FileMan should *not* be altered, per VHA Directive 10-93-142.

The callable entry points for those routines that can be invoked from applications are shown in the "Callable at" column. The callable routines, ScreenMan, and Database Server calls are described in detail (including their function, required variables, and any restrictions) in the *VA FileMan Programmer Manual*.

For information on callable routines, please refer to the "Major APIs" and "Other APIs" sections of the *VA FileMan Programmer Manual*.

For information on ScreenMan, please refer to the "ScreenMan" section of the *VA FileMan Programmer Manual*.

For information on Database Server calls, please refer to the "Database Server (DBS)" chapter in the "Major APIs" section of the *VA FileMan Programmer Manual*.

The Direct mode utilities, which can only be called directly from M and ScreenMan-specific utilities, are listed at the end of this section. Both are also described in the *VA FileMan Programmer Manual*.

Routine	**Callable at**	**Routine Description**
%DT		See DIDT for callable entry points and description.
%DTC		See DIDTC for callable entry points and description.
%RCR		See DIRCR for callable entry points and description.

Routine	**Callable at**	**Routine Description**
DDBR	EN^DDBR WP^DDBR BROWSE^DDBR DOCLIST^DDBR	Routines responsible for displaying ASCII text on a terminal screen, for viewing only.
DDBR0		
DDBR1		
DDBR2		
DDBR3		
DDBR4		
DDBRAHT		
DDBRAHTE		
DDBRAHTJ		
DDBRAHTR		
DDBRAP		
DDBRGE		
DDBRP		
DDBRS		
DDBRT	$$TEST^DDBRT	
DDBRU		
DDBRU2		
DDBRWB		
DDBRZIS	CLOSE^DDBRZIS OPEN^DDBRZIS POST^DDBRZIS	
DDFIX		Routine that checks nodes in the data dictionary and the FILE file (#1).
DDGF		Routines used to create and edit ScreenMan forms.
DDGF0		
DDGF1		
DDGF2		
DDGF3		
DDGF4		
DDGFADL		
DDGFAPC		
DDGFASUB		

Routine	Callable at	Routine Description
DDGFBK DDGFBSEL DDGFEL DDGFFLD DDGFFLDA DDGFFM DDGFH DDGFHBK DDGFLOAD DDGFORD DDGFPG DDGFSV DDGFU DDGFUPDB DDGFUPDP		
DDGLIB0 DDGLIBH DDGLIBW DDGLIBW1		Routines that manage the screen for VA FileMan's screen-oriented utilities.
DDIOL	EN^DDIOL	Routine that either writes text to the screen, writes text in ScreenMan's Command Area, or loads text into an array, depending on the environment in which it is called.
DDMAP DDMAP1 DDMAP2		Routines that generate a graphic display of the pointer relationships among a specified group of package files to an output device.
DDMP DDMP1 DDMP2 DDMPSM DDMPSM1 DDMPU	FILE^DDMP	Routines used by the Import Tool.
DDR DDR0 DDR1 DDR2 DDR3 DDR4		Routines that contain the RPCs for the VA FileMan Delphi components.

Routine	**Callable at**	**Routine Description**
DDS	DDS	Routines used to compile and run forms for data viewing and editing—ScreenMan.
DDS0		
DDS01		
DDS02		
DDS1		
DDS10		
DDS11		
DDS2		
DDS3		
DDS4		
DDS41		
DDS5		
DDS6		
DDS7		
DDSBOX		
DDSCAP		
DDSCLONE		
DDSCLONF		
DDSCOM		
DDSCOMP		
DDSDBLK		
DDSDEL		
DDSDFRM		
DDSFO		
DDSIT		
DDSLIB		
DDSM		
DDSM1		
DDSMSG		
DDSOPT		
DDSPRNT		
DDSPRNT1		
DDSPRNT2		
DDSPTR		
DDSR		
DDSR1		
DDSRSEL		
DDSRUN		
DDSSTK		
DDSU		

Routine	Callable at	Routine Description
DDSUTL	MSG^DDSUTL REFRESH^DDSUTL REQ^DDSUTL UNED^DDSUTL	
DDSVAL	$$GET^DDSVAL PUT^DDSVAL	
DDSVALF	$$GET^DDSVALF PUT^DDSVALF	
DDSVALM		
DDSWP		
DDSZ		
DDSZ1		
DDSZ2		
DDSZ3		
DDU		Routines responsible for running the data dictionary checking utility.
DDUCHK		
DDUCHK1		
DDUCHK2		
DDUCHK3		
DDUCHK4		
DDUCHK5		
DDW		Routines responsible for full screen text editing.
DDW1		
DDW2		
DDW3		
DDW4		
DDW5		
DDW6		
DDW7		
DDW8		
DDW9		
DDWC		
DDWC1		

Routine	**Callable at**	**Routine Description**
DDWF DDWG DDWH DDWK DDWT1		
DDXP DDXP1 DDXP2 DDXP3 DDXP31 DDXP32 DDXP33 DDXP4 DDXP41 DDXP5 DDXPLIB		Routines responsible for the data export to a Foreign Format tool.
DI		Routine for direct entry into VA FileMan.
DIA DIA1 DIA2 DIA3		Routines responsible for gathering fields to be edited.
DIAC	DIAC	Routine that determines file access.
DIALOG	BLD^ DIALOG $$EZBLD^ DIALOG	Routines to build VA FileMan dialogues and their functions.
DIALOGU		
DIAR DIARA DIARB DIARCALC DIARR DIARR1 DIARR2 DIARR3 DIARR4 DIARR5		Routines responsible for VA FileMan archiving.

Routine	**Callable at**	**Routine Description**
DIARR6		
DIARU		
DIARX		
DIAU		Routine used for auditing.
DIAX	EN^DIAXU	Routines responsible for extracting data to a VA FileMan file.
DIAXD		
DIAXERR		
DIAXF		
DIAXM		
DIAXM1		
DIAXM2		
DIAXM3		
DIAXMS		
DIAXP		
DIAXT		
DIAXU		
DIB	EN^DIB	Routine that creates a new file.
DIBT		Routine that stores a SORT template.
DIBT1		
DIBTEDT		
DIC	DIC FIND^DIC $$FIND1^DIC IX^DIC LIST^DIC	Routines that perform VA FileMan lookups or return an ordered list of records.
DIC0		
DIC1	MIX^DIC1 DO^DIC1	
DIC11		
DIC2		
DIC3		
DIC4		
DICA		Routines responsible for DBS Updater functions.
DICA1		
DICA2		
DICA3		

Routine	**Callable at**	**Routine Description**
DICATT		Routines responsible for the Modify File Attributes option.
DICATT0		
DICATT1		
DICATT2		
DICATT22		
DICATT3		
DICATT4		
DICATT5		
DICATT6		
DICATTA		Routine responsible for data dictionary audits.
DICATTD		Routines responsible for Modify File Attributes option in Screen oriented format.
DICATTD0		
DICATTD1		
DICATTD2		
DICATTD3		
DICATTD4		
DICATTD5		
DICATTD6		
DICATTD7		
DICATTD9		
DICATTDD		
DICATTDE		
DICATTDK		
DICATTDM		
DICD	WAIT^DICD	Routine for selecting, displaying, editing, or deleting a cross-reference.
DICE		Routines responsible for creating cross-references.
DICE0		
DICE1		
DICE2		
DICE3		
DICE4		
DICE7		
DICF		Routines responsible for DBS Finder functions.
DICF0		
DICF1		
DICF2		
DICF3		

Routine	**Callable at**	**Routine Description**
DICF4 DICF5 DICFIX DICFIX1		
DICL DICL1 DICL10 DICL2 DICL3 DICLIB DICLIX DICLIX0 DICLIX1		Routines responsible for DBS Lister functions.
DICM DICM0 DICM1 DICM2 DICM3		Routines responsible for performing transforms on the lookup value to attempt to find a match on the lookup index(es). For example, transforms date to internal format.
DICN DICN0 DICN1	FILE^DICN YN^DICN	Routines that allow adding a new entry to a file.
DICOMP DICOMP0 DICOMP1 DICOMPV DICOMPW DICOMPX DICOMPY DICOMPZ		Routines that evaluate computed field expressions.
DICQ DICQ1	DQ^DICQ	Routines responsible for help on lookups.
DICR		Routine responsible for recursive calls for cross-references on triggered fields.
DICRW DICRW1	DT^DICRW	Routines that select a file.

Routine	**Callable at**	**Routine Description**
DICU DICU1 DICU11 DICU2 DICUF DICUIX DICUIX1 DICUIX2		Routines containing utilities used during lookups.
DID	EN^DID FIELD^DID FIELDLST^DID FILE^DID FILELST^DID $$GET1^DID	Routines for data dictionary listings.
DID1		Standard data dictionary listing.
DID2		Modified data dictionary listing.
DIDC		Condensed data dictionary listing.
DIDG		Global Map data dictionary listing.
DIDH		Headers for the data dictionary listings.
DIDH1		
DIDX		Brief data dictionary listing.
DIDT	%DT DD^%DT	Routine responsible for the Date/Time validation. Must be stored in the Manager Account as %DT.
DIDTC	%DTC C^%DTC NOW^%DTC H^%DTC DW^%DTC YMD^%DTC COMMA^%DTC S^%DTC YX^%DTC HELP^%DTC	Routine responsible for the Date/Time operations. Must be stored in the Manager Account as %DTC.

Routine	Callable at	Routine Description
DIDU DIDU1 DIDU2		Routines responsible for data dictionary functions.
DIE DIE0 DIE1 DIE17 DIE2 DIE3 DIE9 DIED DIEF DIEF1 DIEFU DIEFW DIEH DIEH1 DIEKMSG DIEQ DIEQ1	DIE CHK^DIE FILE^DIE HELP^DIE $$KEYVAL^DIE UPDATE^DIE VAL^DIE VALS^DIE WP^DIE	Routines responsible for the Enter or Edit File Entries option and for DBS filing and Help retrieval functions.
DIENV DIENVSTP DIENVWRN		Environment check routines.
DIET DIETED		Routine that displays an INPUT template and performs VA FileMan auditing function.

Routine	**Callable at**	**Routine Description**
DIEV		Routines responsible for data validation functions.
DIEV1		
DIEVK		
DIEVK1		
DIEVS		
DIEZ	DIEZ	Routines that compile INPUT templates.
DIEZ0	EN^DIEZ	
DIEZ1		
DIEZ2		
DIEZ3		
DIEZ4		
DIFG	DIFG	Routines responsible for Filegrams.
DIFG0		
DIFG0A		
DIFG0B		
DIFG1		
DIFG2		
DIFG3		
DIFG3A		
DIFG4		
DIFG4A		
DIFG5		
DIFG6		
DIFG7		
DIFGA		
DIFGA1		
DIFGB		
DIFGG	EN^DIFGG	
DIFGG2		
DIFGG4		
DIFGGI		
DIFGGSB		
DIFGGSB1		
DIFGGSB2		
DIFGGU		
DIFGO		
DIFGSRV		

Routine	Callable at	Routine Description
DIFROM DIFROM0 DIFROM1 DIFROM11 DIFROM12 DIFROM2 DIFROM3 DIFROM4 DIFROM41 DIFROM42 DIFROM5 DIFROM6 DIFROM7 DIFROMH DIFROMH1 DIFROMS DIFROMS1 DIFROMS2 DIFROMS3 DIFROMS4 DIFROMS5 DIFROMSB DIFROMSC DIFROMSD DIFROMSE DIFROMSI DIFROMSK DIFROMSL DIFROMSO DIFROMSP DIFROMSR DIFROMSS DIFROMSU DIFROMSV DIFROMSX DIFROMSY	DIFROM	Routines responsible for generating init packages for export and supporting Kernel's KIDS functions.
DIG		Routine responsible for the Scattergram option on the Statistics submenu.
DIH		Routine responsible for the Histogram option on the Statistics submenu.

Routine	Callable at	Routine Description
DII DII1		Routines responsible for the main menu in stand-alone VA FileMan and for the Inquire to File Entries option.
DIIS DIISS		Routines responsible for device selection for stand-alone VA FileMan. Stored in the Manager Account as %ZIS and %ZISS.
DIK	DIK IXALL^DIK IX^DIK IX1^DIK ENALL^DIK EN^DIK EN1^DIK	Routines that perform file re-indexing and entry deletion.
DIK1		
DIKC DIKC1 DIKC2 DIKCDD DIKCFORM DIKCP DIKCP1 DIKCP2 DIKCP3 DIKCU DIKCU1 DIKCU2 DIKCUTL DIKCUTL1 DIKCUTL2 DIKCUTL3		Routines responsible for defining, printing, and executing the logic for New-Style indices.
DIKK DIKK1 DIKK2 DIKKDD DIKKFORM DIKKP DIKKUTL DIKKUTL1 DIKKUTL2		Routines responsible for defining, printing, and verifying the integrity of Keys.

Routine	Callable at	Routine Description
DIKKUTL3 DIKKUTL4		
DIKZ DIKZ0 DIKZ1 DIKZ11 DIKZ2	DIKZ EN^DIKZ	Routines responsible for VA FileMan's cross-reference compiler.
DIL DIL0 DIL1 DIL11 DIL2 DILL		Routines responsible for processing PRINT templates or fields.
DILF	CLEAN^DILF $$CREF^DILF DA^DILF DT^DILF FDA^DILF $$IENS^DILF $$OREF^DILF $$VALUE1^DILF VALUES^DILF	Routine that contains VA FileMan's library of functions.
DILFD	$$EXTERNAL^DILFD $$FLDNUM^DILFD PRD^DILFD RECALL^DILFD $$ROOT^DILFD $$VFIELD^DILFD $$VFILE^DILFD	

Routine	Callable at	Routine Description
DILIBF		
DIM	DIM	Routines responsible for the M syntax checker.
DIM1		
DIM2		
DIM3		
DIM4		
DINIT		Routines that initialize VA FileMan.
DINIT*		Numerous routines starting with "DINIT" are used in the initialization process.
DINTEG		Routines used for integrity checking of VA FileMan routines.
DINTEG*		Several routines starting with "DINTEG" are used during integrity checking.
DINVDTM		Routines that DINZMGR saves as %ZOSV and, for some operating systems, %ZOSV1, during a stand-alone VA FileMan installation.
DINV1DTM		
DINVMSM		
DINVONT		
DINVVXD		
DINV1VXD		
DINZMGR		Routines that rename routines in the Manager Account and set up ^%ZOSF nodes for stand-alone VA FileMan.
DINZMGR1		
DINZDTM		
DINZMSM		
DINZONT		
DINZVXD		
DIO		Routines responsible for building sort logic, executing the sort, and performing output functions.
DIO0		
DIO1		
DIO2	DT^DIO2	
DIO3		
DIO4		
DIOS		
DIOS1		
DIOC		Routine responsible for checking code to check query conditions.

Routine	**Callable at**	**Routine Description**
DIOQ		Routine responsible for determining sort (query) optimization numbers.
DIOU		Routines responsible for generic VA FileMan code generation utilities.
DIOZ	^DIOZ	Routines responsible for compiling SORT templates.
DIP DIP0 DIP1 DIP10 DIP100 DIP11 DIP12 DIP2 DIP21 DIP22 DIP23 DIP3 DIP31 DIP4 DIP5	EN1^DIP	Routines that: process sorting specifications, edit SORT templates, process the FROM and TO sort range, edit PRINT templates, process PRINT templates, and initialize the printing process.
DIPKINIT		Routines that install a new data dictionary for the PACKAGE file. The data dictionary brings no data.
DIPKI*		There are numerous routines starting with "DIPKI" used in the PACKAGE file initialization. (During VA FileMan Installation, DINIT automatically installs the PACKAGE file at sites not running Kernel V. 8.0 or later.)
DIPT	DIPT DIBT^DIPT	Routine that displays PRINT and SORT templates.
DIPTED		Routine used for the ScreenMan-based PRINT template editor.

Routine	**Callable at**	**Routine Description**
DIPZ	DIPZ EN^DIPZ	Routines that compile PRINT templates.
DIPZ0 DIPZ1 DIPZ2		
DIQ	EN^DIQ Y^DIQ D^DIQ DT^DIQ $$GET1^DIQ GETS^DIQ	Routines that retrieve data and support DBS Retriever and DD Retriever functions.
DIQ1	EN^DIQ1	
DIQG DIQGDD DIQGDD0 DIQGDDF DIQGDDT DIQGDDU DIQGQ DIQGU DIQGU0		
DIQQ DIQQ1 DIQQQ		Routines that provide Help on various subjects.
DIR DIR0 DIR01 DIR02 DIR03 DIR0H DIR0K DIR0W DIR1 DIR2 DIR3 DIRQ	DIR	Routines responsible for the standard reader used in VA FileMan.
DIRCR	XY^%RCR	Routine that moves arrays. Must be stored in the Manager Account as %RCR.

Routine	Callable at	Routine Description
DIS DIS0 DIS1 DIS2 DIS3	EN^DIS	Routines responsible for the Search File Entries option.
DISZ*		Temporary routines compiled for SORT templates and deleted after use (*not* exported with VA FileMan routines).
DIT DIT0 DIT1 DIT2 DIT3 DITP DITR DITR1		Routines responsible for the Transfer Entries option. Also used by the Compare/Merge option and by DIFROM.
DITC DITC0 DITC1 DITC2 DITC3		Routines responsible for allowing a user to select data values during the compare/merge process.
DITM DITM1 DITM2 DITMGM1 DITMGM2 DITMGM2A DITMGM2B DITMGM2C DITMGMRG DITMGMRI DITMU1 DITMU2 DITMU3 DITMU4		Routines used to compare/merge two records located within a single file.
DITP		Routine responsible for transferring pointers.

Routine	**Callable at**	**Routine Description**
DIU DIU0 DIU1 DIU2 DIU20 DIU21 DIU3 DIU31 DIU4 DIU5	EN^DIU2	Routines responsible for the Utility Functions option.
DIUTL		General utility routines used internally by VA FileMan.
DIV DIVC DIVR DIVR1 DIVU		Routines that verify field data.
DIVRE DIVRE1		Routine that checks for required field data.
DIWE DIWE1 DIWE11 DIWE12 DIWE2 DIWE3 DIWE4 DIWE5	EN^DIWE	Routines responsible for VA FileMan's Line Editor and display of word processing output. They also provide for use of Alternate Editors.
DIWF	DIWF EN1^DIWF EN2^DIWF	Routine used for printing forms.
DIWP DIWW	DIWP DIWW	Routines responsible for display of word processing output.

Routine	**Callable at**	**Routine Description**
DIX		Routines used for the Statistics option.
DIXC		Routine used for the Descriptive Statistics option.
DMSQ		Routines used to build and maintain an SQL mapping to VA FileMan data. Allows access to VA FileMan data using an SQL interface.
DMSQD		
DMSQE		
DMSQF		
DMSQF1		
DMSQF2		
DMSQP		
DMSQP1		
DMSQP2		
DMSQP3		
DMSQP4		
DMSQP5		
DMSQP6		
DMSQS		
DMSQT		
DMSQT1		
DMSQU		

Direct Mode Utilities

In addition to the callable entry points shown in the previous section, there are a few other entry points into VA FileMan routines. Unlike the callable entry points, these entries **cannot be used within application programs**. Only users with programmer access can invoke the following direct mode utilities from the M prompt:

```
C^DI
D^DI
P^DI
Q^DI
```

For more information on these direct mode utilities, please refer to the "^DI: Programmer Access" chapter in the "Developer Tools" section of the *VA FileMan Programmer Manual.*

ScreenMan-Specific Utilities

The following are ScreenMan-specific utilities:

```
^DDGF
CLONE^DDS
PRINT^DDS
RESET^DDS
```

For more information on these ScreenMan-specific utilities, please refer to the "Prog. Mode Utilities" topic in the "ScreenMan Forms" chapter in the "ScreenMan" section of the *VA FileMan Programmer Manual.*

File List

This section lists all the VA FileMan files with their file numbers, shows their global location, and gives a brief description. Data exported with stand-alone VA FileMan is shown. VA FileMan uses files numbered between 0 and 2. VA FileMan files should not be altered, per VHA Directive 10-93-142.

.11 INDEX **Global Location: ^DD("IX",**

The INDEX file stores information about New-Style cross-references defined on a file. Whereas Traditional cross-references are stored under the 1 nodes of the ^DD for a particular field, New-Style cross-references are stored in this file and can consist of one field (simple cross-references), as well as more than one field (compound cross-references).

.2 DESTINATION **Global Location: ^DIC(.2**

The DESTINATION file documents the location where data is used.

.31 KEY **Global Location: ^DD("KEY",**

The KEY file stores information about keys on a file or subfile. A key is a set of one or more fields that uniquely identifies a record in a file. If more than one set of fields can uniquely identify a record, one of those sets should be designated the primary key; all others should be designated secondary keys. The primary key is the principal means of identifying records in the file. To allow VA FileMan to enforce key uniqueness, the database designer must define a regular index that consists of all the fields that make up the key. This index is called the uniqueness index. All key fields must have values. They cannot be null.

.4 PRINT TEMPLATE **Global Location: ^DIPT(**

The PRINT TEMPLATE file stores VA FileMan PRINT templates. Exported PRINT templates include: CAPTIONED, FILE SECURITY CODES, DI-PKG-DEFAULT-DEFINITION, DDXP FORMAT DOC, and DDXP FORMAT DOC HDR.

.401 SORT TEMPLATE   Global Location: ^DIBT(

The SORT TEMPLATE file stores VA FileMan SORT, SEARCH, and INQUIRE templates.

.402 INPUT TEMPLATE   Global Location: ^DIE(

The INPUT TEMPLATE file stores VA FileMan INPUT templates.

.403 FORM   Global Location: ^DIST(.403

The FORM file stores forms used by VA FileMan to display screens. The DDXP FF FORM1 and various forms used by ScreenMan's Form Editor utility are exported.

.404 BLOCK   Global Location: ^DIST(.404

The BLOCK file stores blocks used to build forms for screen display. Blocks are exported for use with the forms sent with VA FileMan.

.44 FOREIGN FORMAT   Global Location: ^DIST(.44

The FOREIGN FORMAT file holds specifications for sending data to an application outside of M. Several Foreign Formats are exported.

.46 IMPORT TEMPLATE   Global Location   ^DIST(.46,

The IMPORT TEMPLATE file holds specifications for importing information from an application outside of M into a VA FileMan file.

.5 FUNCTION   Global Location: ^DD("FUNC"

The FUNCTION file stores the computed functions available in VA FileMan. The functions described in the *VA FileMan Advanced User Manual* are exported.

ℹ For more information on functions, please refer to the "VA FileMan Functions" chapter in the "Tools" section of the *VA FileMan Advanced User Manual*.

.6 DD AUDIT Global Location: ^DDA(

The DD AUDIT file stores the changes made to data dictionaries.

.7 MUMPS OPERATING SYSTEM Global Location: ^DD("OS"

The MUMPS OPERATING SYSTEM file stores the operating systems recognized by VA FileMan along with operating system-specific data. This data is exported.

.81 DATA TYPE Global Location: ^DI(.81

The DATA TYPE file stores information about the DATA TYPEs known to VA FileMan. Several DATA TYPEs are exported.

.83 COMPILED ROUTINE Global Location: ^DI(.83

The COMPILED ROUTINE file contains a list of numbers (to be used to create compiled Sort routines) and a flag to indicate whether a number is currently in use.

.84 DIALOG Global Location: ^DI(.84

The DIALOG file contains text used to "talk" to the user (error messages, help text, prompts). Entries under IEN 10,000 are exported by VA FileMan and are used in VA FileMan routines.

.85 LANGUAGE Global Location: ^DI(.85

The LANGUAGE file is used to reference subentries in the DIALOG file for user dialogue in foreign languages and contains M code used to perform data transformations for such things as dates and numbers to non-English formats.

1 FILE Global Location: ^DIC(

The FILE file stores the name, number, global name or location, package name, security access, and developer of VA FileMan created files. Data for the VA FileMan files is exported.

1.1 AUDIT Global Location: ^DIA(

The AUDIT file stores the date and time, user's name, and old and new data values of changes made to audited fields.

1.11 ARCHIVAL ACTIVITY Global Location: ^DIAR(1.11

The ARCHIVAL ACTIVITY file stores information about and status of archiving and extract activities.

1.12 FILEGRAM HISTORY Global Location: ^DIAR(1.12

The FILEGRAM HISTORY file stores information and status of Filegrams.

1.13 FILEGRAM ERROR LOG Global Location: ^DIAR(1.13

The FILEGRAM ERROR LOG file stores information about Filegram errors and the text of the affected Filegram.

1.2 ALTERNATE EDITOR Global Location: ^DIST(1.2

The ALTERNATE EDITOR file stores information about the editors that can be used to edit VA FileMan's WORD-PROCESSING-type fields. Data for the Line Editor and the Screen Editor is exported.

1.521 SQLI_SCHEMA Global Location: ^DMSQ("S",

The SQLI_SCHEMA file stores a set of tables and domains; a subset of catalog and environment.

1.52101 SQLI_KEY_WORD Global Location: ^DMSQ("K",

The SQLI_KEY_WORD file stores the SQL identifiers that *cannot* be used for column and table names. SQL, ODBC, and vendors all have lists of restricted words, which should be put in this table before SQLI table generation.

1.5211 SQLI_DATA_TYPE Global Location: ^DMSQ("DT",

The SQLI_DATA_TYPE file stores a set of values from which all domains of that type can be drawn:

- PRIMARY_KEY—The set of all primary keys (in SQLI_TABLE_ELEMENT file, type P).
- CHARACTER—The set of all character strings of length less than 256.
- INTEGER—The set of all cardinal numbers.
- NUMERIC—The set of all real numbers.
- DATE—The set of all date valued tokens.
- TIME—The set of all time valued tokens.
- MOMENT—The set of all tokens which have both a date and a time value.
- BOOLEAN—The set of all tokens which evaluate to true or false only.
- MEMO—The set of all character strings of length greater than 255.

1.5212 SQLI_DOMAIN Global Location: ^DMSQ("DM",

The SQLI_DOMAIN file stores the set from which all objects of that domain must be drawn. In SQLI, all table elements (SQLI_TABLE_ELEMENT file) have a domain that restricts them to their domain set. For each DATA TYPE there is a domain of the same name, representing the same set. Other domains have different set membership restrictions.

Each domain has a DATA TYPE, which determines the rules for comparing values from different domains, and the operators that can be used on them.

The PRIMARY_KEY DATA TYPE and domain is unique to SQLI. It is used to relate primary keys to foreign keys unambiguously.

For information on table elements, see the SQLI_TABLE_ELEMENT file below.

1.5213 SQLI_KEY_FORMAT Global Location: ^DMSQ("KF",

The SQLI_KEY_FORMAT file stores strategies for converting base values into key values. Soundex and uppercase conversion are common examples. This implies that comparisons of key values with base values must be preceded by conversion of the base value to a key value. Key formats are frequently lossy; they can't be converted uniquely back to base format.

1.5214 SQLI_OUTPUT_FORMAT Global Location: ^DMSQ("OF",

The SQLI_OUTPUT_FORMAT file stores strategies for converting base values to external values. In VA FileMan, they are used to convert references to pointers to their text values. They are also used for the SET OF CODES type.

SQLI projects POINTER TO A FILE and SET OF CODES as calls to $$GET1^DIQ, VARIABLE-POINTERs into calls to $$EXTERNAL^DILFD.

Vendors and other users of SQLI can implement their own conversions to improve performance.

1.5215 SQLI_TABLE Global Location: ^DMSQ("T",

The SQLI_TABLE file stores the descriptor of a set of table elements: includes name and file number (see the SQLI_TABLE_ELEMENT file below). Each ^DD(DA) represents a table in a relational model of VA FileMan. Further, each index represents a table.

Each schema contains multiple tables. Each table contains just one primary key, but multiple columns, foreign keys and indices.

1.5216 SQLI_TABLE_ELEMENT Global Location: ^DMSQ("E",

The SQLI_TABLE_ELEMENT file contains the names and domains of primary keys, columns, and foreign keys. Each represents the relational concept of an attribute, whose essential characteristics are a name (unique by relation) and a domain.

For more information, see the SQLI_PRIMARY_KEY, SQLI_COLUMN, and SQLI_FOREIGN KEY files.

1.5217 SQLI_COLUMN Global Location: ^DMSQ("C",

The SQLI_COLUMN file stores a set of formatting and physical structure specifications. Each column specification has a column type table element (SQLI_TABLE_ELEMENT file) that contains the relational specifications, name, and domain. The column specification contains those attributes required to locate the value in the global structure and to project the value to the user.

For information on table elements, see the SQLI_TABLE_ELEMENT file above.

1.5218 SQLI_PRIMARY_KEY Global Location: ^DMSQ("P",

The SQLI_PRIMARY_KEY file stores a chosen set of columns that uniquely identify a table. In the relational model (as in set theory) the columns of a primary key are not ordered. In SQLI, they must be, in order to map to the quasi-hierarchical model of M globals.

VA FileMan subfiles (multiples) have a primary key element for each parent plus one for the subfile. Each contains a pointer to its primary key table element (SQLI_TABLE-ELEMENT file), a sequence, and a column in the local base table (SQLI_COLUMN file).

For information, see the SQLI_TABLE_ELEMENT and SQLI_COLUMN files above.

1.5219 SQLI_FOREIGN_KEY Global Location: ^DMSQ("F",

The SQLI_FOREIGN_KEY file stores a set of columns in a table that match the primary key of another table. They represent an explicit join of the two tables. Each foreign key element points to its table element (SQLI_TABLE_ELEMENT file), a column in the local table (SQLI_COLUMN file), and a primary key element of a foreign table (SQLI_PRIMARY_KEY file). The primary key table element of the foreign table has the domain of that table, which makes the connection.

For more information, see the SQLI_TABLE_ELEMENT, SQLI_COLUMN, and SQLI_PRIMARY_KEY files.

1.52191 SQLI_ERROR_TEXT Global Location: ^DMSQ("ET",

The SQLI_ERROR_TEXT file stores a numbered list of error messages, auto-generated by ERR^DMSQU.

1.52192 SQLI_ERROR_LOG Global Location: ^DMSQ("EX",

The SQLI_ERROR_LOG file stores a log of all errors encountered while compiling SQLI. It generates the error text table (SQLI_ERROR_TEXT file) on a LAYGO basis; errors are added only when they occur. If DBS errors triggered the error, the DIALOG file reference is also saved.

ℹ For more information, see the SQLI_ERROR_TEXT and DIALOG files above.

The DINIT routines install the files listed above.

Another set of init routines (DIPKINIT), called by DINIT during installation, is sent with each release of VA FileMan. These routines will install the PACKAGE file (#9.4), if you are running a version of Kernel prior to Version 8.0 or if you are running stand-alone VA FileMan without Kernel. The PACKAGE file is necessary to build inits using DIFROM. A single entry for DIPK is created in the PACKAGE file by the DIPKINITs.

ℹ For more information on DIFROM, please refer to the "DIFROM" chapter in the "Developer's Tools" section of the *VA FileMan Programmer Manual.*

> ⚠ **The Kernel Installation and Distribution System (KIDS) has replaced the use of DIFROM as the method of exporting software packages in the VA. The version of DIFROM released with VA FileMan V. 22.0 does *not* transport the new Key and Index structures and should *not* be used to transport any file making use of these new features.**

Exported Options

Stand-alone VA FileMan

The exported menu structure of VA FileMan is displayed below. The options are the ones provided to stand-alone VA FileMan users.

ENTER OR EDIT FILE ENTRIES

PRINT FILE ENTRIES

SEARCH FILE ENTRIES

MODIFY FILE ATTRIBUTES

INQUIRE TO FILE ENTRIES

UTILITY FUNCTIONS

- VERIFY FIELDS
- CROSS-REFERENCE A FIELD OR FILE
- IDENTIFIER
- RE-INDEX FILE
- INPUT TRANSFORM (SYNTAX)
- EDIT FILE
- OUTPUT TRANSFORM
- TEMPLATE EDIT
- UNEDITABLE DATA
- MANDATORY/REQUIRED FIELD CHECK
- KEY DEFINITION

OTHER FILEGRAMS OPTIONS

- CREATE/EDIT FILEGRAM TEMPLATE
- DISPLAY FILEGRAM TEMPLATE
- GENERATE FILEGRAM
- VIEW FILEGRAM
- SPECIFIERS
- INSTALL/VERIFY FILEGRAM

ARCHIVING	SELECT ENTRIES TO ARCHIVE ADD/DELETE SELECTED ENTRIES PRINT SELECTED ENTRIES CREATE FILEGRAM ARCHIVING TEMPLATE WRITE ENTRIES TO TEMPORARY STORAGE MOVE ARCHIVED DATA TO PERMANENT STORAGE PURGE STORED ENTRIES CANCEL ARCHIVAL SELECTION FIND ARCHIVED ENTRIES
AUDITING	FIELDS BEING AUDITED DATA DICTIONARIES BEING AUDITED PURGE DATA AUDITS PURGE DD AUDITS TURN DATA AUDIT ON/OFF
SCREENMAN	EDIT/CREATE A FORM RUN A FORM DELETE A FORM PURGE UNUSED BLOCKS
STATISTICS	DESCRIPTIVE STATISTICS SCATTERGRAM HISTOGRAM

- EXTRACT DATA TO FILEMAN FILE
    - SELECT ENTRIES TO EXTRACT
    - ADD/DELETE SELECTED ENTRIES
    - PRINT SELECTED ENTRIES
    - MODIFY DESTINATION FILE
    - CREATE EXTRACT TEMPLATE
    - UPDATE DESTINATION FILE
    - PURGE EXTRACTED ENTRIES
    - CANCEL EXTRACT SELECTION
    - VALIDATE EXTRACT TEMPLATE
- DATA EXPORT TO FOREIGN FORMAT
    - DEFINE FOREIGN FILE FORMAT
    - SELECT FIELDS FOR EXPORT
    - CREATE EXPORT TEMPLATE
    - EXPORT DATA
    - PRINT FORMAT DOCUMENTATION
- IMPORT DATA
- BROWSER

DATA DICTIONARY UTILITIES

- LIST FILE ATTRIBUTES
- MAP POINTER RELATIONS
- CHECK/FIX DD STRUCTURE

TRANSFER ENTRIES

- TRANSFER FILE ENTRIES
- COMPARE/MERGE FILE ENTRIES

VA FileMan with Kernel

VA FileMan exports the options presented in the charts that follow. They are installed if Kernel exists on your system. The top-level VA FileMan menu option, DIUSER, can be found on Kernel's EVE menu. The top-level menu option, DMSQ MENU, will *not* be attached to any other existing menu; it is stand-alone, and can be assigned as needed.

```
VA FileMan (DIUSER)
|
|
|-------------------------------------------------------- Enter or Edit File Entries
|                                                         [DIEDIT]
|
|-------------------------------------------------------- Print File Entries
|                                                         [DIPRINT]
|
|-------------------------------------------------------- Search File Entries
|                                                         [DISEARCH]
|
|-------------------------------------------------------- Modify File Attributes
|                                                         [DIMODIFY]
|
|-------------------------------------------------------- Inquire to File Entries
|                                                         [DIINQUIRE]
|
|---- Utility Functions---------------------------------- Verify Fields
|     [DIUTILITY]                                         [DIVERIFY]
|         |
|         |---------------------------------------------- Cross-Reference A Field
|         |                                               [DIXREF]
|         |
|         |---------------------------------------------- Identifier
|         |                                               [DIIDENT]
|         |
|         |---------------------------------------------- Re-Index File
|         |                                               [DIRDEX]
|         |
|         |---------------------------------------------- Input Transform (Syntax)
|         |                                               [DIITRAN]
|         |
|         |---------------------------------------------- Edit File
|         |                                               [DIEDFILE]
|         |
|         |---------------------------------------------- Output Transform
|         |                                               [DIOTRAN]
|         |
|         |---------------------------------------------- Template Edit
|         |                                               [DITEMP]
|         |
|         |---------------------------------------------- Uneditable Data
|         |                                               [DIUNEDIT]
|         |
|         |---------------------------------------------- Mandatory/Required
|         |                                               Field Check
|         |                                               [DIFIELD CHECK]
|         |
|         |---------------------------------------------- Key Definition
|                                                         [DIKEY]
```

```
|---- Data Dictionary Utilities-------------------------- List File Attributes
|     [DIDDU]                                             [DILIST]
|         |
|         |---------------------------------------------- Map Pointer Relations
|         |                                               [DI DDMAP]
|         |
|         |---------------------------------------------- Check/Fix DD Structure
|                                                         [DI DDUCHK]
|
|
|-------------------------------------------------------- Transfer Entries
|                                                         [DITRANSFER]
|
|---- Other Options ----------- Filegrams --------------- Create/Edit Filegram
      [DIOTHER]                 [DIFG]                    Template
          |                     **LOCKED: XUFILEGRAM**    [DIFG CREATE]
          |                         |                     **LOCKED: XUFILEGRAM**
          |                         |
          |                         |-------------------- Display Filegram Template
          |                         |                     [DIFG DISPLAY]
          |                         |                     **LOCKED: XUFILEGRAM**
          |                         |
          |                         |-------------------- Generate Filegram
          |                         |                     [DIFG GENERATE]
          |                         |                     **LOCKED: XUFILEGRAM**
          |                         |
          |                         |-------------------- View Filegram
          |                         |                     [DIFG VIEW]
          |                         |
          |                         |-------------------- Specifiers
          |                         |                     [DIFG SPECIFIERS]
          |                         |                     **LOCKED: XUFILEGRAM**
          |                         |
          |                         |-------------------- Install/Verify Filegram
          |                                               [DIFG INSTALL]
          |                                               **LOCKED: XUFILEGRAM**
          |
          |
          |-------------------- Audit Menu -------------- Fields Being Audited
          |                     [DIAUDIT]                 [DIAUDITED FIELDS]
          |                     **LOCKED: XUAUDITING**
          |
          |                         |
          |                         |-------------------- Data Dictionaries
          |                         |                     Being Audited
          |                         |                     [DIAUDIT DD]
          |                         |
          |                         |-------------------- Purge Data Audits
          |                         |                     [DIAUDIT PURGE DATA]
          |                         |
          |                         |-------------------- Purge DD Audits
          |                         |                     [DIAUDIT PURGE DD]
          |                         |
          |                         |-------------------- Turn Data Audit On/Off
          |                                               [DIAUDIT TURN ON/OFF]
          |
          |
          |-------------------- ScreenMan  -------------- Edit Create A Form
          |                     [DDS SCREEN MENU]         [DDS EDIT/CREATE A FORM]
          |                     **LOCKED: XUSCREENMAN**
          |                         |
          |                         |-------------------- Run a Form
          |                         |                     [DDSRUN A FORM]
          |                         |
          |                         |-------------------- Delete a Form
          |                         |                     [DDS DELETE A FORM]
          |                         |
          |                         |-------------------- Purge Unused Blocks
          |                                               [DDS PURGE UNUSED BLOCKS]
          |
          |
          |---------------------------------------------- Statistics
          |                                               [DISTATISTICS]
          |
          |-------------------- VA FileMan -------------- Data Dictionary
          |                     Management                Cross-reference
          |                     [DI MGMT MENU]            Compile/Uncompile
          |                     **LOCKED: XUMGR**         [DI DD COMPILE]
          |                         |
          |                         |-------------------- Input Template
          |                         |                     Compile/Uncompile
          |                         |                     [DI INPUT COMPILE]
          |                         |
          |                         |-------------------- Print Template
          |                         |                     Compile/Uncompile
          |                         |                     [DI PRINT COMPILE]
          |                         |
          |                         |-------------------- Sort Template
          |                         |                     Compile/Uncompile
          |                         |                     [DI SORT COMPILE]
          |                         |
          |                         |-------------------- Re-Initialize VA FileMan
          |                         |                     [DI REINITIALIZE]
          |                         |
          |                         |-------------------- Set Type of Mumps Operating
          |                         |                     System
          |                         |                     [DI SET MUMPS OS]
          |                         |
          |                         |-------------------- Forms Print
          |                                               [DIWF]
          |
          |
          |-------------------- Data Export to ---------- Define Foreign File Format
          |                     Foreign Format            [DDXP DEFINE FORMAT]
          |                     [DDXP EXPORT MENU]        **LOCKED: DDXP-DEFINE**
          |                         |
          |                         |
          |                         |-------------------- Select Fields for Export
          |                         |                     [DDXP SELECT EXPORT FIELDS]
          |                         |
          |                         |-------------------- Create Export Template
          |                         |                     [DDXP CREATE EXPORT
          |                         |                     TEMPLATE]
          |                         |
          |                         |-------------------- Export Data
          |                         |                     [DDXP EXPORT DATA]
          |                         |
          |                         |-------------------- Print Format Documentation
          |                                               [DDXP FORMAT DOCUMENTATION]
```

```
|-------------------- Extract Data To --------- Select Entries to Extract
|                     Fileman File              [DIAX SELECT]
|                     [DIAX EXTRACT MENU]       **LOCKED: DIEXTRACT**
|                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Add/Delete Selected Entries
|                         |                     [DIAX ADD/DELETE]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Print Selected Entries
|                         |                     [DIAX PRINT]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Modify Destination File
|                         |                     [DIAX MODIFY]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Create Extract Template
|                         |                     [DIAX CREATE]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Update Destination File
|                         |                     [DIAX UPDATE]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Cancel Extract
|                         |                     Selection [DIAX CANCEL]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Purge Extracted
|                         |                     Entries  [DIAX PURGE]
|                         |                     **LOCKED: DIEXTRACT**
|                         |
|                         |-------------------- Validate Extract
|                                               Template [DIAX VALIDATE]
|                                               **LOCKED: DIEXTRACT**
|
|
|-----------------------------------------------Import Data
|                                               [DDMP IMPORT]
|
|---------------------------------------------- Browser
                                                [DDBROWSER]
```

```
SQLI (VA FileMan) (DMSQ MENU)
|
|----------------------------------------- RUN Regenerate SQLI Projection
|                                              [DMSQ PROJECT]
|                                              **LOCKED: XUPROGMODE**
|
|----------------------------------------- WHY Find Out SQLI Status
|                                              [DMSQ DIAGNOSTICS]
|
|----------------------------------------- ERR Print Errors from Last Projection
|                                              [DMSQ PRINT ERRORS]
|
|----------------------------------------- X    Purge SQLI Data [DMSQ PURGE]
|                                               **LOCKED: XUPROGMODE**
|
|---- DD Table Statistics Reports -------- DD1  Field Listing by File (Brief)
|     [DMSQTS MENU]                             [DMSQ TS FIELDS BRIEF]
|         |
|         |------------------------------- DD2  Field Listing by File (Full)
|         |                                     [DMSQ TS FIELDS FULL]
|         |
|         |------------------------------- IN1  List Subfile Links (Brief)
|         |                                     [DMSQ TS SUBFILE BRIEF]
|         |
|         |------------------------------- IN2  List Incoming Pointer/Subfile Links
|         |                                     (Full)
|         |                                     [DMSQ TS PTR SUBFILE FULL]
|         |
|         |------------------------------- OUT1 List Pointer and Parent Links (Brief)
|         |                                     [DMSQ TS PTR PARENT BRIEF]
|         |
|         |------------------------------- OUT2 List Pointer and Parent Links (Full)
|         |                                     [DMSQ TS PTR PARENT FULL]
|         |
|         |------------------------------- CNT1 Pointer Statistics by Individual Table
|         |                                     [DMSQ TS PTR STATS]
|         |
|         |------------------------------- CNT2 Pointer Statistics (Summary)
|         |                                     [DMSQ TS PTR STATS SUMMARY]
|         |
|         |------------------------------- NAME Table Name Listing (VA FileMan vs.
|                                               SQLI)
|                                               [DMSQ TS NAMES]
|
|---- CNTS Site Statistics Reports ------- TBL  Table Total (Excluding Index Tables)
|     [DMSQPS MENU]                             [DMSQ PS TOTAL TABLES]
|         |
|         |------------------------------- 1C   Column Total (All Tables)
|         |                                     [DMSQ PS TOTAL COLUMNS]
|         |
|         |------------------------------- INDX Index Table Total
|         |                                     [DMSQ PS TOTAL INDEXES]
|         |
|         |------------------------------- ELEM Table Element Totals, By Type
|         |                                     [DMSQ PS TOTAL TABLE ELEMENTS]
|         |
|         |------------------------------- 2C   Column Totals, by Table
|         |                                     [DMSQ PS TOTAL TABLE COLS]
|         |
|         |------------------------------- 3C   Column Totals, by Table
|         |                                     (Ordered by # of Columns)
|         |                                     [DMSQ PS TOTAL TABLE COLS A]
|         |
|         |------------------------------- 4C   Columns in Regular Tables Total
|         |                                     [DMSQ PS TOTAL COLUMNS REG]
|         |
|         |------------------------------- FLDS Columns in Regular Tables, Excluding
|         |                                     ID Columns
|         |                                     [DMSQ PS COLUMNS REG NOID]
|         |
|         |------------------------------- DOM  Columns by Domain
|                                               [DMSQ PS COLUMNS BY DOMAIN]
|
|----------------------------------------- GRP  Suggest Table Groupings
                                                [DMSQ SUGGEST TABLE GROUPINGS]
```

Cross-references

This section contains a description of the MUMPS-type cross-references that exist on fields in VA FileMan files. There are no bulletin or trigger cross-references in these files. All other cross-references are regular ones used for lookup or sorting, or both.

The cross-references are grouped by file. The field affected is identified along with the cross-reference's name (or subscript location if there is no name) and a brief description. Many of these cross-references are described in more detail in the data dictionaries.

PRINT TEMPLATE File (#.4)

Field	X-Ref ID	Description
NAME	F_file#	This cross-reference is used to quickly find all PRINT templates associated with a particular file.
	AF	This cross-reference sets up an "AF" cross-reference node for each field in a compiled PRINT template. The cross-reference has the form: `^DIPT("AF",file#,field#,print template#)=""`
FILE	F_file#	This cross-reference is used to quickly find all PRINT templates associated with a particular file.
TEMPLATE TYPE	FG	This cross-reference is used to do a quick lookup of FILEGRAM-type of PRINT templates.
	EX	This cross-reference is used to do a quick lookup of EXTRACT-type PRINT templates.

Table 2: PRINT TEMPLATE file Cross-references

SORT TEMPLATE File (#.401)

Field	X-Ref ID	Description
NAME	F_file#	This cross-reference is used to quickly find all SORT templates associated with a particular file.
FILE	F_file#	This cross-reference is used to quickly find all SORT templates associated with a particular file.

Table 3: SORT TEMPLATE file Cross-references

INPUT TEMPLATE File (#.402)

Field	X-Ref ID	Description
NAME	F_file#	This cross-reference is used to quickly find all INPUT templates associated with a particular file.
	AF	This cross-reference sets up an "AF" cross-reference node for each field in a compiled INPUT template. The cross-reference has the form: `^DIE("AF",file#,field#,input template#)=""`
FILE	F_file#	This cross-reference is used to quickly find all INPUT templates associated with a particular file.

Table 4: INPUT TEMPLATE file Cross-references

FORM File (#.403)

Field	X-Ref ID	Description
NAME	F1 AY	This cross-reference is used to quickly find all ScreenMan forms associated with a particular file. This cross-reference merely documents the existence of data stored under ^DIST(.403,form IEN,"AY"). This is where the compiled data for a form is stored.
PAGE NAME (subfield of PAGE multiple)	C	This cross-reference stores the PAGE NAME converted to uppercase characters.
PRIMARY FILE	F	This cross-reference is used to quickly find all ScreenMan forms associated with a particular file.
IS THIS A POP UP PAGE? (subfield of PAGE multiple)	AC	This cross-reference ensures that no header block, next page, or previous page is associated with a pop up page.

Table 5: FORM file Cross-references

BLOCK File (#.404)

Field	X-Ref ID	Description
CAPTION (subfield of FIELD multiple)	C	This cross-reference is used for lookup of fields by CAPTION. It is also used for ^-jumping.
UNIQUE NAME (subfield of FIELD multiple)	D	This cross-reference stores the UNIQUE NAME converted to uppercase characters.

Table 6: BLOCK file Cross-references

IMPORT TEMPLATE File (#.46)

Field	X-Ref ID	Description
NAME	F1	Creates an index under F_file# that is used for lookup when the file number is known.
PRIMARY FILE	F	Same as F1.

Table 7: IMPORT TEMPLATE file Cross-references

FILE of Files (#1)

Field	X-Ref ID	Description
NAME	AD	This cross-reference sets and kills the "GL" node for the file. This node has the form: `^DIC(file#,0,"GL")=file's global location`
	AE	This cross-reference sets and kills the "NM" node for the file. This node has the form: `^DIC(file#,0,"NM")=file's name`

Table 8: FILE of files Cross-references

SQLI_TABLE_ELEMENT File (#1.5216)

Field	X-Ref ID	Description
E_TABLE	G	Table element by table, by name.
E_TYPE	F	Table element by table, by type.

Table 9: SQLI_TABLE_ELEMENT file Cross-references

SQLI_COLUMN File (#1.5217)

Field	X-Ref ID	Description
C_FIELD	D	Column by VA FileMan file number, by field number.

Table 10: SQLI_COLUMN file Cross-references

SQLI_PRIMARY_KEY File (#1.5218)

Field	X-Ref ID	Description
P_SEQUENCE	C	Primary key by table, by sequence.

Table 11: SQLI_PRIMARY_KEY file Cross-references

Archiving and Purging

Archiving

There are no package-specific archiving procedures in VA FileMan.

The generic archiving tool for **V*IST*A** is a part of VA FileMan. It is described in the *VA FileMan Advanced User Manual*.

For more information on archiving, please refer to the "Archiving" section of the *VA FileMan Advanced User Manual*.

The Extract Tool provides a means of archiving data into a VA FileMan file. It is also described in the *VA FileMan Advanced User Manual*.

For more information on the Extract Tool, please refer to the "Extract Tool" chapter in the "Archiving" section of the *VA FileMan Advanced User Manual*.

Purging

Within VA FileMan, the only files that might grow large enough to require purging of data are the audit files. The AUDIT (#1.1) and DD AUDIT (#.6) files capture information about changes to data and to data dictionaries, respectively. These audits are started and stopped by using the options on the Auditing submenu. The amount of data accumulated is dependent both on the scope of the audit and its duration. Options are available to purge the AUDIT file (Purge Data Audits) and the DD AUDIT file (Purge DD Audits). Purging the audit files is optional. Decisions to purge must be made based on the size of the files and any need to retain the audit data.

For instructions on the use of the Auditing options, please refer to the "Auditing" section of the *VA FileMan Advanced User Manual*.

Purge Stored Entries is an option on the Archiving submenu. This option removes the data archived from the primary file and from the ARCHIVAL ACTIVITY file when the archiving process is complete. The Purge Stored Entries option should be run when each archiving action is finished in order to remove the archived data and clean up the files.

On the Extract Tool submenu, the Purge Extracted Entries option removes extracted data from the primary file and from the ARCHIVAL ACTIVITY file when the extract process is complete. This option should be run when using the Extract Tool for archiving purposes to remove extracted data.

External Relations

VA FileMan is designed to function as a stand-alone database management system. It can accomplish most of the activities described in the Programmer and User Manuals without relying on other software (except, of course, an implementation of ANSI Standard M on which to run).

When installing VA FileMan, you identify the current M operating system. An entry in the MUMPS OPERATING SYSTEM file is selected. The information in this file allows things to be accomplished that are done differently by various implementations of M. In addition, when the DINZMGR routine is run during a stand-alone installation, %ZOSF global nodes are set in the Manager Account. These nodes contain additional implementation-specific information that allows for expanded functionality.

For details of installing stand-alone VA FileMan, please refer to the "Stand-alone VA FileMan Installation" topic in the *VA FileMan Installation Guide*.

By attending to at least minimal operating system concerns, VA FileMan can be installed without the remaining parts of Kernel except for the PACKAGE file (see below). However, the capability of VA FileMan is enhanced when it is installed with Kernel and MailMan.

Specifically, VA FileMan V. 22.0 is designed to work with Kernel V. 8.0 or later. For example, user security (via the NEW PERSON file [#200]), control of file access, more sophisticated menu presentation, device control, and queuing are all made possible. Bulletins, one of VA FileMan's cross-references, become operational when MailMan is installed to deliver the messages. The Filegram options also require MailMan.

Kernel allows networking two CPUs with different operating systems (e.g., DSM for OpenVMS and MSM). Kernel provides this ability by retrieving the type of operating system from ^("OS"). This global does not have to be replicated or translated; thus, a separate copy of the global can be stored on each CPU. When running stand-alone VA FileMan, the type of operating system is retrieved either from the second piece of ^%ZOSF("OS"), if the DINZMGR was run, or from ^DD("OS"). ^DD("OS") is the global location of the MUMPS OPERATING SYSTEM file (#.7). The global ^DD must always be either replicated or translated across systems. In any case, the local variable DISYS is used by VA FileMan to store the value of the current operating system. VA FileMan finds some operating system-specific code in nodes descending from ^DD("OS",DISYS); other code is found in ^%ZOSF nodes.

The PACKAGE file (#9.4) is not a VA FileMan file; however, it is shipped with VA FileMan, and is installed automatically during the installation of VA FileMan, if either Kernel is not present or if a version of Kernel prior to V 8.0 is present. The PACKAGE file must be present on your system to use the DIFROM routines to export software packages.

The Kernel Installation and Distribution System (KIDS) has replaced the use of DIFROM as the method of exporting software packages in the VA. The version of DIFROM released with VA FileMan V. 22.0 does *not* transport the new Key and Index structures and should *not* be used to transport any file making use of these new features.

VA FileMan exports options and security keys with the DI and DD namespace for use by Kernel.

Throughout the VA FileMan manuals, specific reference is made to Kernel or MailMan when either is needed for a function to work.

DBA Approvals and Database Integration Agreements (DBIAS)

To obtain the current list of active DBIAs of which VA FileMan is a custodian:

1. Sign on to the **FORUM** system.
2. Select the **DBA menu**.
3. Select the **Integration Agreements Menu**.
4. Select the **Custodial Package Menu**.
5. Choose the **ACTIVE by Custodial Package option**.
6. When this option prompts you for a package, enter **VA FILEMAN**.
7. All current DBIAs for which the VA FileMan package is custodian are listed.

To obtain detailed information on a specific integration agreement:

1. Sign on to the **FORUM** system.
2. Select the **DBA menu**.
3. Select the **Integration Agreements Menu**.
4. Choose the **Inquire option**.
5. When prompted with "Select INTEGRATION REFERENCES:", enter the **integration agreement number of the DBIA you would like to display**.
6. The full text of the requested DBIA will be displayed.

To obtain the current list of DBIAs that VA FileMan is a subscriber to:

1. Sign on to the **FORUM** system.
2. Select the **DBA menu**.
3. Select the **Integration Agreements Menu**.
4. Select the **Subscriber Package Menu**.
5. Choose the **Print ACTIVE by Subscribing Package option**.
6. When prompted with "START WITH SUBSCRIBING PACKAGE: FIRST//", enter **VA FILEMAN** (in uppercase). Then, when prompted with "GO TO SUBSCRIBING PACKAGE: LAST//", enter **VA FILEMAN** (in uppercase).
7. All current DBIAs to which the VA FileMan package is a subscriber are listed.

Internal Relations

All options can be independently invoked. None of the options require any special setup in order to run successfully.

Package-wide Variables

VA FileMan package-wide or key variables that can be assumed to be defined at all times are the following:

DUZ The internal entry number from the NEW PERSON file (#200).

DUZ(0) The variable defining the user's access.

DUZ("LANG") If running Kernel V. 8.0 or later, this variable refers to the language of the current user.

DT The current date in VA FileMan internal format.

DTIME The integer value of the number of seconds the user has to respond to a timed read.

U The up-arrow.

In addition, the following variable has a special meaning for VA FileMan although it will not always be defined:

DISYS The current M operating system—pointer to the MUMPS OPERATING SYSTEM file contained in the first piece of ^DD("OS") and, if using Kernel, in the second piece of ^%ZOSF("OS").

Standards and Conventions (SAC) Exemptions

Beginning January 1, 1995, VA FileMan V. 22.0 has been granted exemptions from the following standards by the Programming Standards and Conventions Committee.

- STANDARD SECTION: 4B **Package-wide variables**

  Beginning December 22, 1994, VA FileMan is exempted from killing the listed variables in the following calls:

Supported Reference	**Variables**
DIC	DA
FILE^DICN	DA
DIE	%,D,D0,DI,DQ,X,D1,%X,%Y
DIK	%,DA,DIC, X, Y
EN1^DIP	X
EN^DIQ1	%,D0,I,J,X,Y,C

- STANDARD SECTION: 6D **FM compatibility**

  - The following globals are exempt from VA FileMan compatibility:

    ^DISV
    ^DOSV

  - VA FileMan may set a non-VA FileMan compatible node [e.g., ^XXX(File#, IEN,-9)] to record information about archival activity and may set non-VA FileMan compatible nodes ^(3) and ^(2) to store old and new values of any audited field.

Pointer Map

The following pages contain a diagram of the pointer relationships between fields in VA FileMan's files. The map reflects the relationships that will exist in a VA FileMan environment that runs Kernel V. 8.0. As files are added to a system, new pointer relationships can be created; thus, the actual map for an operational system may differ.

This diagram was created using the Map Pointer Relations option on the Data Dictionary Utilities submenu.

For more information about creating and reading this map, please refer to the "Map Pointer Relations option" topic in the "List File Attributes" chapter in the "File Management" section of the *VA FileMan Advanced Users Manual.*

```
    File/Package:                          Date: SEP 8,1994

  FILE (#)                  POINTER           (#) FILE
   POINTER FIELD             TYPE           POINTER FIELD    FILE POINTED TO
--------------------------------------------------------------------------------

          L=Laygo      S=File not in set      N=Normal Ref.      C=Xref.
          *=Truncated      m=Multiple           v=Variable Pointer

                                        ---------------------
  KEY (#.31)                            |                   |
       UNIQUENESS INDEX .....  (N )->  |  .11 INDEX        |
                                        ---------------------

                                       ---------------------
ARCHIVAL ACTIVITY (#1.11)              |                   |
   PRINT TEMPLATE .......  (N )->      |  .4 PRINT TEMPLA* |
FILEGRAM HISTORY (#1.12)               |                   |
   FILEGRAM ............   (N )->      |   FILE            |-> FILE
                                       |   DESTINATION FI* |-> FILE
                                       ---------------------

                                        ---------------------
  ARCHIVAL ACTIVITY (#1.11)             |                   |
    SEARCH TEMPLATE ......  (N L)->     |  .401 SORT TEMPL* |
                                        |   FILE            |-> FILE
                                        ---------------------
                                        ---------------------
  KERNEL SITE PARAMETE (#4.3)           |                   |
    USER CHARACTERISTICS T* (N S )->    |  .402 INPUT TEMP* |
  KERNEL SYSTEM PARAME (#8989.3)        |                   |
    USER CHARACTERISTICS T* (N S )->    |   FILE            |-> FILE
                                        ---------------------
```

Pointer Map (continued):

```
                                      ---------------------
FORM (#.4031)                         |                   |
  PAGE:HEADER BLOCK ....  (N L)->     |  .404 BLOCK       |
  PAGE:BLOCK:BLOCK NAME   (N C L)->   |                   |
                                      ---------------------
                                      ---------------------
PRINT TEMPLATE (#.4)                  |                   |
  EXPORT FORMAT ........  (N )->      |  .44 FOREIGN FOR* |
                                      ---------------------
                                      ---------------------
                                      |   .46 IMPORT TEMP* |
                                      |   PRIMARY FILE     |-> FILE
                                      |   CREATOR          |-> NEW PERSON
                                      |   IMPORT:FILE*     |-> FILE
                                      ---------------------
                                      ---------------------
                                      |  .6 DD AUDIT      |
                                      |   USER            |-> NEW PERSON
                                      ---------------------
                                      ---------------------
SORT TEMPLATE (#.4014)                |                   |
  SORT FIEL:DATA TYPE F*  (N )->      |  .81 DATA TYPE    |
PRINT TEMPLATE (#.42)                 |                   |
  EXPORT FIELD:DATA TYPE  (N )->      |                   |
                                      ---------------------
                                      ---------------------
SQLI_ERROR_LOG (#1.52192)             |  .84 DIALOG       |
  FILEMAN_ERROR …….       (N C)->     |   PACKAGE         |-> PACKAGE
                                      ---------------------
                                      ---------------------
DIALOG (#.847)                        |                   |
  TRANSLATION:LANGUAGE .  (N C )->    |  .85 LANGUAGE     |
NEW PERSON (#200)                     |                   |
  LANGUAGE .............  (N S )->    |                   |
KERNEL SITE PARAMETE (#8989.3)        |                   |
  DEFAULT LANGUAGE .....  (N S )->    |                   |
                                      ---------------------
```

Pointer Map (continued):

```
    File/Package(Cont):                          Date: SEP 8,1994

  FILE (#)                  POINTER           (#) FILE
   POINTER FIELD             TYPE           POINTER FIELD    FILE POINTED TO
--------------------------------------------------------------------------------

          L=Laygo      S=File not in set      N=Normal Ref.      C=Xref.
          *=Truncated      m=Multiple           v=Variable Pointer

                                        ---------------------
  VARIABLE-POINTER (#.12)               |                   |
    . . . . . . . . . . .   (N S)->     |                   |
  PRINT TEMPLATE (#.4)                  |   1 FILE          |
    FILE .................  (N )->      |                   |
    DESTINATION FILE ... .  (N )->      |                   |
  SORT TEMPLATE (#.401)                 |   DEVELOPER       |-> NEW PERSON
    FILE .................  (N )->      |                   |
  INPUT TEMPLATE (#.402)                |                   |
    FILE .................  (N )->      |                   |
  IMPORT TEMPLATE  (#.46)               |                   |
    PRIMARY FILE .........  (N )->      |                   |
    IMPORT FIELDS:FILE ...  (N )->      |                   |
  ARCHIVAL ACTIVITY (#1.11)             |                   |
    FILE .................  (N )->      |                   |
    DESTINATION FILE .....  (N )->      |                   |
  FILEGRAM HISTORY (#1.12)              |                   |
    FILE .................  (N )->      |                   |
  PACKAGE (#9.402)                      |                   |
    AFFECTS R:FILE AFFECT*  (N S C )->  |                   |
    *FILE ................  (N S )->    |                   |
    *PRINT TEMPLATE:FILE..  (N S )->    |                   |
    *INPUT TEMPLATE:FILE..  (N S )->    |                   |
    *SORT TEMPLATE:FILE ..  (N S )->    |                   |
    *SCREEN TE:FILE* .....  (N S )->    |                   |
  BUILD (#9.64)                         |                   |
    FILE .................  (N S )->    |                   |
    BUILD COM:BUILD COMPO*  (N S )->    |                   |
    BUILD:ENTRIES:FILE* ..  (N S )->    |                   |
  INSTALL (#9.714)                      |                   |
    FILE .................  (N S C )->  |                   |
    BUILD COM:BUILD COMPO*  (N S C )->  |                   |
  DUPLICATE RESOLUTION (#15.1)          |                   |
    FILE TO BE CHECKED ...  (N S C )->  |                   |
    DUPLICATE:FILE FOR IN*  (N S C )->  |                   |
    DINUM FIL:DINUM FILE *  (N S C )->  |                   |
  NEW PERSON (#200.032)                 |                   |
    ACCESSIBLE FILE ......  (N S C )->  |                   |
  LOCAL KEYWORD (#8984.1)               |                   |
    ASSOCIATED FILE ......  (N S C )->  |                   |
  LOCAL SYNONYM (#8984.3)               |                   |
    ASSOCIATED FILE ......  (N S C )->  |                   |
  LOCAL LOOKUP (#8984.4)                |                   |
    NAME .................  (N S C )->  |                   |
                                        ---------------------


                                        ---------------------
                                        |  1.1 AUDIT        |
                                        |   USER            |-> NEW PERSON
                                        |   MENU OPTION US* |-> OPTION
                                        | v PROTOCOL or OP* |-> OPTION
                                        |                   |-> PROTOCOL
                                        ---------------------
```

Pointer Map (continued):

```
    File/Package(Cont):                          Date: SEP 8,1994

  FILE (#)                  POINTER           (#) FILE
   POINTER FIELD             TYPE           POINTER FIELD    FILE POINTED TO
--------------------------------------------------------------------------------

          L=Laygo      S=File not in set      N=Normal Ref.      C=Xref.
          *=Truncated      m=Multiple           v=Variable Pointer


                                        ---------------------
                                        |  1.11 ARCHIVAL A* |
                                        |   FILE            |-> FILE
                                        |   ARCHIVER        |-> NEW PERSON
                                        |   SELECTOR        |-> NEW PERSON
                                        |   PURGER          |-> NEW PERSON
                                        |   USER PERFORMIN* |-> NEW PERSON
                                        |   DESTINATION FI* |-> FILE
                                        ---------------------
                                        ---------------------
                                        |  1.12 FILEGRAM H* |
                                        |   FILE            |-> FILE
                                        |   MESSAGE         |-> MESSAGE
                                        ---------------------
                                        ---------------------
  NEW PERSON (#200)                     |                   |
    PREFERRED EDITOR ....  (N S ) ->    |  1.2 ALTERNATE E* |
                                        ---------------------
                                        ---------------------
  SQLI_TABLE (#1.5215)                  |                   |
    T_SCHEMA .............  (N L)->     |  1.521 SQLI_SCHE* |
                                        ---------------------
                                        ---------------------
  SQLI_DOMAIN (#1.5212)                 |                   |
    DM_DATA_TYPE .........  (N C )->    |  1.5211 SQLI_DAT* |
  SQLI_KEY_FORMAT (#1.5213)             |                   |
    KF_DATA_TYPE .........  (N C )->    |   D_OUTPUT_FORMAT |->SQLI_OUTPUT_FO*
  SQLI_OUTPUT_FORMAT (#1.5214)          |                   |
    OF_DATA_TYPE .........  (N )->      |                   |
                                        ---------------------
                                        ---------------------
  SQLI_TABLE_ELEMENT (#1.5216)          |                   |
    E_DOMAIN .............  (N C )->    |  1.5212 SQLI_DOM* |
                                        |   DM_DATA_TYPE    |-> SQLI_DATA_TYPE
                                        |   DM_TABLE        |-> SQLI_TABLE
                                        |   DM_OUTPUT_FORM* |->SQLI_OUTPUT_FO*
                                        ---------------------
                                        ---------------------
  SQLI_PRIMARY_KEY (#1.5218)            |                   |
    P_KEY_FORMAT .........  (N )->      |  1.5213 SQLI_KEY* |
                                        |   KF_DATA_TYPE    |-> SQLI_DATA_TYPE
                                        ---------------------
                                        ---------------------
  SQLI_DATA_TYPE (#1.5211)              |                   |
    D_OUTPUT_FORMAT ......  (N )->      |  1.5214 SQLI_OUT* |
  SQLI_DOMAIN (#1.5212)                 |                   |
    DM_OUTPUT_FORMAT .....  (N )->      |   OF_DATA_TYPE    |-> SQLI_DATA_TYPE
  SQLI_COLUMN (#1.5217)                 |                   |
    C_OUTPUT_FORMAT ......  (N C )->    |                   |
                                        ---------------------
```

Pointer Map (continued):

```
                                      ---------------------
SQLI_DOMAIN (#1.5212)                 |                   |
  DM_TABLE .............  (N C )->    |  1.5215 SQLI_TAB* |
SQLI_TABLE (#1.5215)                  |                   |
  T_MASTER_TABLE .......  (N C )->    |   T_SCHEMA        |-> SQLI_SCHEMA
SQLI_TABLE_ELEMENT (#1.5216)          |                   |
  E_TABLE ..............  (N C )->    |   T_MASTER_TABLE  |-> SQLI_TABLE
                                      ---------------------
                                      ---------------------
SQLI_COLUMN (#1.5217)                 |                   |
  C_TABLE_ELEMENT ......  (N C )->    |  1.5216 SQLI_TAB* |
SQLI_PRIMARY_KEY (#1.5218)            |                   |
  P_TBL_ELEMENT ........  (N C )->    |   E_DOMAIN        |-> SQLI_DOMAIN
SQLI_FOREIGN_KEY (#1.5219)            |                   |
  F_TBL_ELEMENT ........  (N C )->    |   E_TABLE         |-> SQLI_TABLE
                                      ---------------------
                                      ---------------------
SQLI_COLUMN (#1.5217)                 |                   |
  C_PARENT .............  (N C )->    |  1.5217 SQLI_COL* |
SQLI_PRIMARY_KEY (#1.5218)            |                   |
  P_COLUMN .............  (N C )->    |   C_TABLE_ELEMENT |->SQLI_TABLE_ELE*
SQLI_FOREIGN_KEY (#1.5219)            |                   |
  F_CLM_ELEMENT ........  (N )->      |   C_PARENT        |-> SQLI_COLUMN
                                      |   C_OUTPUT_FORMAT |->SQLI_OUTPUT_FO*
                                      ---------------------
                                      ---------------------
SQLI_FOREIGN_KEY (#1.5219)            |                   |
  F_PK_ELEMENT .........  (N )->      |  1.5218 SQLI_PRI* |
                                      |   P_TBL_ELEMENT   |->SQLI_TABLE_ELE*
                                      |   P_COLUMN        |-> SQLI_COLUMN
                                      |   P_KEY_FORMAT    |->SQLI_KEY_FORMAT
                                      ---------------------
                                      ---------------------
                                      |  1.5219 SQLI_FOR* |
                                      |   F_TBL_ELEMENT   |->SQLI_TABLE_ELE*
                                      |   F_PK_ELEMENT    |->SQLI_PRIMARY_K*
                                      |   F_CLM_ELEMENT   |-> SQLI_COLUMN
                                      ---------------------
                                      ---------------------
SQLI_ERROR_LOG (#1.52192)             |                   |
  ERROR ................  (N C L)->   |  1.52191 SQLI_ER* |
                                      ---------------------
                                      ---------------------
                                      |  1.52192 SQLI_ER* |
                                      |   ERROR           |-> SQLI_ERROR_TE*
                                      ---------------------
```

Globals

VA FileMan's globals are listed below:

^DD
^DDA
^DI
^DIA
^DIAR
^DIBT
^DIC
^DIE
^DIPT
^DIST
^DISV
^DIZ
^DMSQ
^DOPT
^DOSV
^TMP
^UTILITY
^%ZOSF

The ^UTILITYand ^TMP globals are temporary globals used and then killed by many VA FileMan options. The ^%ZOSF global is used by the Screen Editor and ScreenMan. If VA FileMan is used with Kernel, nodes in ^%ZOSF are set up during Kernel's installation.

For details about setting up the ^%ZOSF nodes at stand-alone sites, please refer to the *VA FileMan Installation Guide*.

There is a supported entry point to the ^DD global: ^DD("DD"). Its use is explained in the "X ^DD("DD")—Another Way to Convert Dates" topic in the "Date/Time Utilities" section found in the "Classic FileMan" chapter (listed by category) in the "Major APIs" section of the *VA FileMan Programmer Manual*.

For specific information on ^%DT, please refer to the "^%DT" topic in the "Classic FileMan API" chapter in the "Major APIs" section of the *VA FileMan Programmer Manual*.

^DD("VERSION") can be read to get the version number of the VA FileMan package that exists in the system.

Global Journaling, Translation, and Replication

This section contains recommendations for journaling, translating, and replicating VA FileMan globals. Translation is called "impliciting" when running M/SQL. Sites using MSM should see VA's "486 Cookbook" for current recommendations; do *not* use the information here.

Journaling

Journaling is recommended for the following globals:

^DD
^DI
^DIBT
^DIC
^DIE
^DIPT
^DIST
^DIZ

The ^DIZ global is the default location for locally-produced files. Management of this global should be decided locally.

Do *not* journal (translation is optional) the ^DISV global.

Translation

Translation is recommended for the following globals:

^DDA
^DIA
^DIBT
^DIC
^DIZ

The ^DIZ global is the default location for locally-produced files. Management of this global should be decided locally.

Replication

Replication is recommended for the following globals:

^DD
^DI
^DIAR
^DIE
^DIPT
^DIST
^DMSQ
^DOPT

Mapping Routines (DSM for OpenVMS)

Routine mapping is at the discretion of the systems manager. The RTHIST routines provide a method for each site to determine the extent to which certain routines are utilized.

The following list is provided as a *recommendation only*:

In the **MGR Account**:

%DT, %DTC, %RCR

In the **VAH Account**:

DDBR*
DDIOL
DDS, DDS0*, DDS1*, DDSCOM, DDSR*, DDSU
DDW4, DDW5, DDW6, DDWT1
DIAC, DIALOG*
DIC, DIC0, DIC1*, DIC2, DIC3
DICA, DICA1, DICA2, DICA3
DICD
DICF*
DICL*
DICM*
DICN*
DICQ*
DICR, DICRW
DICU*
DIE, DIE0, DIE1, DIE2, DIE3, DIE9, DIE17
DIED
DIEF*
DIEH*
DIEQ*
DIEV*
DIK, DIK1, DIKC, DIKC1, DIKC2, DIKCU, DIKCU1, DIKCU2
DIL*
DIO*
DIP, DIP0, DIP1*, DIP2, DIP21, DIP22, DIP3*, DIP4, DIP5
DIQG, DIQGQ, DIQGU*
DIW*
DIR, DIR0, DIR01, DIR1, DIR2, DIR3, DIRQ

⚠ **DIEZ*, DIFRO*, DIKZ*, DIOZ*, DIPZ* must *not* be mapped.**

Software Product Security

VA FileMan is the database management system for **V***IST***A**. As such, it provides security on a file, field, and template level. This security is based on a string of characters stored in the local variable DUZ(0). You can find the details of the data security system imposed by VA FileMan in the *VA FileMan Advanced User Manual.* The security mechanisms described there apply to the files and data sent with the VA FileMan package as well as to the files created by other application packages and by users.

For specific information on VA FileMan's data security, please refer to the "Data Security" chapter in the "Security" section of the *VA FileMan Advanced User Manual*.

When used with Kernel, other types of access control are available. Kernel's Sign-on/Security component is described in the *Kernel Systems Manual*. If Kernel's File Access Security system has been implemented on your system, you can use it to control user access to files.

When you use VA FileMan within the Kernel's menu system, you are subject to the Kernel's security requirements:

- You must enter correct Access and Verify Codes.
- You can only use menus and options to which you have been granted access.
- You must have the proper keys to use certain locked options.

The VA FileMan menus are found on the DIUSER option. This option is usually located on the EVE menu option distributed with Kernel.

To see a diagram of the complete menu tree for VA FileMan, please refer to the "Exported Options" section in this manual.

Security Management

There are *no special legal requirements* involved in the use of the VA FileMan product, nor does VA FileMan relieve users of any legal requirements.

Mail Groups and Alerts

VA FileMan does *not* make use of mail groups or alerts.

Remote Systems

VA FileMan does *not* transmit data to any remote system, facility, or database.

Interfacing

No non-VA products are embedded in or required by VA FileMan, other than those provided by the underlying operating systems.

Electronic Signatures

Electronic signatures are *not* used within VA FileMan.

Security Keys

VA FileMan options are locked with the security keys described below. The keys in the XU namespace are distributed by Kernel; however, they lock VA FileMan options. The two remaining keys are distributed by VA FileMan and are installed when DINIT is run:

XUAUDITING This key is needed to access the Auditing menu or to run any of the Auditing options.

XUFILEGRAM This key is needed to access the Filegram menu or to run any of the Filegram options except the View Filegram option, for which no key is required.

XUMGR This key is for users who act as site management staff. It is required in order to access the VA FileMan Management menu. It is also needed to access many Kernel options.

XUPROGMODE This key is needed to access the SQLI Regenerate SQLI Projection and Purge SQLI Data options.

XUSCREENMAN This key is needed to access the ScreenMan menu.

DDXP-DEFINE This key is needed to access the Export Tool's Define Foreign File Format option.

DIEXTRACT This key is needed to access the Extract Data to FileMan File menu.

File Security

Files with numbers less than two (2) belong to VA FileMan. In general, these files *cannot* be directly accessed. You can access them only through the menu options. Those users who are granted programmer access (DUZ(0)="@") can directly read and manipulate data in VA FileMan files. However, it is *strongly recommended* that changes to data in such files only be made through documented VA FileMan utilities.

References

The following directive specifies that VA FileMan routines and files should *not* be altered:

VHA Directive 10-93-142

Official Policies

Modification of any part of the VA FileMan software is *not permitted* as per VHA Directive 10-93-142.

Distribution of the VA FileMan software is unrestricted.

Glossary

ANSI Standard MUMPS — **A**merican **N**ational **S**tandards **I**nstitute (ANSI) computer language used by VA FileMan. Also called **M**. The acronym MUMPS stands for **M**assachusetts General Hospital **U**tility **M**ulti**p**rogramming **S**ystem.

Archiving — The storing of historical or little used data offline (often on tape).

Auditing — The monitoring and recording of computer use. VA FileMan audits can log changes to data values in files and to the structure of the file itself.

Browser — An interactive application in VA FileMan that displays ASCII text on a terminal that supports a scroll region. The text can be in the form of a VA FileMan WORD-PROCESSING-type field or sequential local or global array. The user is allowed to navigate freely within the document.

Callable Entry Points — Places in a VA FileMan routine that can be called from an application program.

Checksum Value — A number computed for each routine in a package. The number is used to verify that the routine is uncorrupted and unchanged. Any coding change to a routine will change its checksum value.

Cross-reference — In VA FileMan, an attribute of a field that identifies an action to take place when the value of the field is changed. Often, the action is the placement of the field's value into an index. In Version 22.0 of VA FileMan, the INDEX file allows creation of indexes that contain more than one data field. Thus, they become an attribute of the file, rather than of a single field. The action described in the INDEX file entry happens when any of the involved fields is changed.

Data Dictionary — A data dictionary (DD) contains the definitions of a file's elements (fields or data attributes), relationships to other files, and structure or design.

Database Management System A collection of software that handles the storage, retrieval and updating of records in a database.

DBS **D**ata**b**ase **S**erver: an Application Programmer Interface (API) for VA FileMan that updates the database in a non-interactive mode. VA FileMan passes information that needs to be displayed to the user to the calling routine in arrays.

DBMS **D**ata**b**ase **M**anagement **S**ystem.

Device A terminal, printer, modem or other type of hardware or equipment associated with a computer. A Host file of an underlying operating system may be treated like a device in that it can be written to (e.g., for spooling).

DHCP The **D**ecentralized **H**ospital **C**omputer **P**rogram, see "**V**_IST_**A**."

Direct Mode Utility An entry point into a routine that can only be called from programmer mode, see "Callable Entry Points."

DSM for OpenVMS The current name for VAX DSM(V6). One of the M operating systems supported by VA FileMan.

Entry For VA FileMan, an instance of a file; a set of logically related data in a file; a record.

Field In an entry, a specified area used for the value of a data attribute. The data specifications of each VA FileMan field are documented in the file's data dictionary.

File A set of related records (or entries) treated as a unit.

Filegrams A VA FileMan feature that stores file information in a sequential format in preparation for archiving or for sending it to a corresponding database in another computing location.

Global In M, global may refer to a variable stored on disk ("global variable") or the array to which the global variable may belong ("global array").

Help Frames Online screens of documentation made possible by the Kernel's Help Processor.

Impliciting Term used by M/SQL operating system for global translation.

Init A step in the installation process that builds VA FileMan files from a set of routines (the "init routines"). Shortened form for "initialization."

Index A part of the data global whose subscripts are one or more fields from a single record in the file, along with the internal entry number (or numbers) that locate the record. An ordered list of all or a subset of the records in the file used to facilitate lookup and sorting.

INDEX File This file was introduced with Version 22.0 of VA FileMan. Contains the information that describes an index on a file. Old-style index information is stored descendent from the description of the indexed field in the data dictionary. The INDEX file allows the creation of more complex indexes.

Journaling The capturing of changes to files in order to facilitate the restoring of files from a known prior state.

Kernel A set of **V***IST***A** software utilities that function as an intermediary between the host operating system and **V***IST***A** application packages (e.g., Laboratory, Pharmacy, IFCAP, etc.). Kernel provides a standard and consistent user and programmer interface between application packages and the underlying M implementation.

Key A group of one or more fields that together uniquely identifies a record in a file. Each key field must have a value, and fields that make up a key must in combination be unique for all records in the file. VA FileMan enforces key integrity.

Key Variable See "Package-wide Variable" below.

LAYGO Access A user's authorization to create a new entry when editing a computer file. **L**earn **A**s **Y**ou **GO**: the ability to create new entries.

MailMan An electronic mail system (e-mail) that allows you to send messages to and receive them from other users via the computer. It is part of **V**_IST_**A**.

Mapping See "Routine Mapping."

Operating System A basic program that runs on the computer, controls the peripherals, allocates computing time to each user, and communicates with terminals. Some M implementations take over the functions of an operating system completely; others run on top of another host operating system.

Package The set of programs, files, documentation, online Help, and installation procedures required for a given software application package identified by a unique namespace. Elements include routines, files, and file entries from the OPTION, KEY, HELP FRAME, BULLETIN, FUNCTION, SORT TEMPLATE, PRINT TEMPLATE, INPUT TEMPLATE, FORM, and BLOCK files. Packages are transported using VA FileMan's DIFROM routine, which creates initialization (init) routines to bundle the files and entries for export.

Package-wide Variable For **V**_IST_**A**, a variable that, for a particular application package, has a standard and documented meaning. Some package-wide variables may need to be defined at all times during package use. Also called Key Variable.

Pointer Relationships In VA FileMan, links between files that are created by use of the POINTER TO A FILE or VARIABLE-POINTER DATA TYPEs.

Programmer Access The ability to utilize VA FileMan features that are reserved for application developers. Referred to as "having the at-sign (@)" because "@" is the DUZ(0) value that grants programmer access.

Programmer Mode Entry into VA FileMan directly from the M prompt instead of from Kernel's menu system (e.g., by entering **D P^DI** at the M prompt).

Replication (of globals) The practice of keeping and maintaining identical copies of the same global in different physical locations.

Routine A program or a sequence of instructions called by a program that may have some general or frequent use. M routines are groups of program lines that are saved, loaded, and called as a single unit via a specific name.

Routine Mapping The placement of routines into main memory. Frequently used routines are mapped to reduce disk access and thereby increase efficiency.

SAC Exemption An exception specifically granted by the **S**tandards **a**nd **C**onventions Committee of the Programming Standards and Conventions requirements.

ScreenMan A VA FileMan screen-oriented utility that supports creation, alteration, and presentation of screens for data editing and data display.

SDP space **S**equential **D**isk **P**rocessor space is an area on disk set aside for temporary storage of data during copying of the data. SDP is implemented by some M systems (e.g., DSM for OpenVMS).

Spacebar Return The use of the key combination <SPACE><RET> at a prompt. VA FileMan retrieves the user's last response to that prompt.

Stand-alone Referring to VA FileMan, the use of VA FileMan without the complete Kernel. The rest of Kernel adds functionality; however, VA FileMan can be used alone.

Template A means of storing report formats, data entry formats, and sorted entry sequences. A template is a permanent place to store selected field specifications for use at a later time.

Translation (of globals) The pointing to a physical disk storage location in another UCI for location of a global. Allows the same globals to be accessed from multiple UCIs.

V*IST***A** The **V**eterans Health **I**nformation **S**ystems and **T**echnology **A**rchitecture, within the Department of Veterans Affairs, is the component of the Veterans Health Administration that develops software and installs, maintains, and updates compatible computer systems in VA medical facilities. (Previously known as the Decentralized Hospital Computer Program [DHCP].)

Index