

**VA FileMan 22.2 and Kernel 8.0**  
**File Access Security User Guide**



**July 2007**

**Revised July 2025**

**Department of Veterans Affairs (VA)**  
**Office of Information and Technology (OIT)**  
**Product Delivery Service (PDS)**

## Revision History

Date	Revision	Description	Author
07/30/2025	2.0	Updates: <ul style="list-style-type: none"><li>• Remodeled document to follow current documentation standards and style guidelines.</li><li>• Content updated based on review of the "File Access Security" section in the <a href="#">Kernel 8.0 Systems Management: Signon/Security User Guide</a>.</li></ul>	Vista Application Shared Services (VASS) Development Team
07/--/2007	1.0	Initial release of <i>VA FileMan 22.2 and Kernel 8.0 File Access Security User Guide</i> .	VASS Development Team

# Table of Contents

Revision History .....	ii
List of Figures.....	iii
List of Tables .....	iii
Orientation .....	iv
<b>1 Introduction.....</b>	<b>1</b>
1.1 VA FileMan Reserved Symbols Used to Implement File Access Security .....	1
<b>2 What Type of File Access Security is Your Site Using? .....</b>	<b>3</b>
<b>3 How to Use the File Access Security at Your Site.....</b>	<b>6</b>
3.1 Classic VA FileMan File Access Security .....	6
3.1.1 Kernel File Access Security vs. Classic VA FileMan File Access Security.....	7
3.1.2 File Level Security Properties .....	7
3.2 Kernel File Access Security.....	10
3.2.1 File Level Security Properties .....	11

## List of Figures

Figure 1: Example—Verify Site is Using Kernel File Access Security .....	4
Figure 2: Example—Security Property Values in the NEW PERSON (#200) File .....	5

## List of Tables

Table 1: Documentation Symbol Descriptions .....	v
Table 2: File Level Security Properties—Classic VA FileMan File Access Security .....	7

## Orientation

### How to Use this Manual

Throughout this manual, advice and instruction are offered about VA FileMan and Kernel 8.0 file access security for Veterans Health Information Systems and Technology Architecture (VistA) system management and application developers.

### Intended Audience

The intended audience of this manual is the following stakeholders:

- System Administrators—System administrators at Department of Veterans Affairs (VA) sites who are responsible for computer management and system security on the VistA M Servers.
- Product Delivery Service (PDS)—VistA legacy development teams.
- Information Security Officers (ISOs)—Personnel at VA sites responsible for system security.
- Product Support (PS).

### Disclaimers

#### Software Disclaimer

This software was developed at the Department of Veterans Affairs (VA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17 Section 105 of the United States Code this software is *not* subject to copyright protection and is in the public domain. VA assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used. This software can be redistributed freely provided that any derivative works bear some notice that they are derived from it.



**CAUTION: Kernel routines should never be modified at the site. If there is an immediate national requirement, the changes should be made by emergency Kernel patch. Kernel software is subject to FDA regulations requiring Blood Bank Review, among other limitations. Line 3 of all Kernel routines states:**

**Per VHA Directive 2004-038, this routine should not be modified.**



**CAUTION:** To protect the security of VistA systems, distribution of this software for use on any other computer system by VistA sites is prohibited. All requests for copies of Kernel for *non-VistA* use should be referred to the VistA site's local Office of Information and Technology Field Office (OITFO).

## Documentation Disclaimers

The appearance of external hyperlink references in this manual does *not* constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA does *not* exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of the VA.

## Documentation Conventions

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

- Various symbols are used throughout the documentation to alert the reader to special information. [Table 1](#) gives a description of each of these symbols:

**Table 1: Documentation Symbol Descriptions**

Symbol	Description
	<b>NOTE/REF:</b> Used to inform the reader of general information including references to additional reading material.
	<b>CAUTION/DISCLAIMER:</b> Used to caution the reader to take special notice of critical information.

- Descriptive text is presented in a proportional font (as represented by this font).
- Conventions for displaying TEST data in this document are as follows:
  - The first three digits (prefix) of any Social Security Numbers (SSN) will begin with either "000" or "666".
  - Patient and user names are formatted as follows:  
<Application Name/Abbreviation/Namespace>PATIENT,<N>  
<Application Name/Abbreviation/Namespace>USER,<N>

Where:

<Application Name/Abbreviation/Namespace> is defined in the Approved Application Abbreviations document.

<N> represents the first name as a number spelled out and incremented with each new entry.

For example, in VA FileMan (DI or FM) or Kernel (XU or KRN) test patient and user names would be documented as follows:

FMPATIENT,ONE; KRNPATIENT,ONE; KRNPATIENT,TWO;  
KRNPATIENT,THREE; ... KRNPATIENT,14; and so on.

FMUSER,ONE; KRNUSER,ONE; KRNUSER,TWO; KRNUSER,THREE; ...  
KRNUSER,14; and so on.

- “Snapshots” of computer commands and online displays (that is screen captures/dialogs) and computer source code, if any, are shown in a *non*-proportional font and may be enclosed within a box.
  - User’s responses to online prompts will be **bold** typeface and highlighted in yellow (such as **<Enter>**).
  - Emphasis within a dialog box will be highlighted in blue (such as **STANDARD LISTENER: RUNNING**).
  - Some software code reserved/key words will be **bold** typeface with alternate color font.
  - References to “**<Enter>**” within these snapshots indicate that the user should press the **<Enter>** key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing **<PF1>**.
  - Author’s comments are displayed in italics or as “callout” boxes.



**NOTE:** Callout boxes refer to labels or descriptions usually enclosed within a box, which point to specific areas of a displayed image.

- This manual refers to the M programming language. Under the 1995 American National Standards Institute (ANSI) standard, M is the primary name of the MUMPS programming language, and MUMPS will be considered an alternate name. This manual uses the name M.

- Descriptions of direct mode utilities are prefaced with the standard M ">" prompt to emphasize that the call is to be used *only in direct mode*. They also include the M command used to invoke the utility. The following is an example:

>D ^XUP

- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field/file names, and security keys (such as the **XUPROGMODE** security key).



**NOTE:** Other software code (such as Delphi/Pascal and Java) variable names and file/folder names can be written in lower or mixed case.

## Internal Word Navigation Links Setup Steps

This document uses Microsoft® Word's built-in navigation for internal hyperlinks. To add **Back** and **Forward** navigation buttons to your toolbar for all Word documents, see the *Tech Writer Tips: Internal Word Navigation Links Setup* document.



**NOTE:** This is a one-time setup and is automatically available in any other Word document once you install it on the Toolbar.

## How to Obtain Technical Information Online

Exported VistA M Server-based software file, routine, and global documentation can be generated using Kernel, MailMan, and VA FileMan utilities.



**NOTE:** Methods of obtaining specific technical information online will be indicated where applicable under the appropriate section.

## Help at Prompts

VistA M Server-based software provides online help and commonly used system default prompts. Users are encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of VistA M Server-based software.

## Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the **List File Attributes** [DILIST] option on the **Data Dictionary Utilities** [DI DDU] menu in VA FileMan to print formatted data dictionaries.



**REF:** For details about obtaining data dictionaries and about the formats available, see the “List File Attributes” chapter in the “File Management” section of the *VA FileMan Advanced User Manual*.

## Assumptions

This manual is written with the assumption that the reader is familiar with the following:

- VistA computing environment:
  - Kernel—VistA M Server software
  - VA FileMan data structures and terminology—VistA M Server software
- Microsoft® Windows environment
- M programming language

## Reference Materials

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

- VA FileMan User Guide
- VA FileMan Advanced User Guide
- VA FileMan Developer’s Guide
- VA FileMan Technical Manual
- Kernel 8.0 Systems Management Guide: Main Directory
- Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide
- Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual
- Kernel Security Tools Manual

VistA documentation is made available online in Microsoft® Word format and in Adobe® Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe® Acrobat Reader, which is freely distributed by Adobe® Systems Incorporated at the following website: [Adobe Website](#).

VistA documentation can be downloaded from the VA Software Document Library (VDL) Website: [VDL Website](#).

VistA documentation and software can also be downloaded from the Network File Share (NFS) repository.

# 1 Introduction

VA FileMan is VistA's database management system. VA FileMan APIs are divided up into two categories:

- **Classic VA FileMan**—Certain modules within VA FileMan are callable by other M routines or when using the VA FileMan exported menu options. File Access Security checking is performed via Classic VA FileMan APIs.



**NOTE:** Another implementation of File Access Security checking is done via Kernel File Access Security. This topic is detailed further in the "File Access Security" section in the "File Access Security" section in the [Kernel 8.0 Systems Management: Signon/Security User Guide](#).

- **Database Server (DBS)**—No File Access Security checking is performed on VA FileMan Database Server (DBS) calls, because they separate interactions with the database created by developers. They are "silent," meaning there is no interaction with the end-user.



**NOTE:** There is no further discussion about the VA FileMan Database Server (DBS) calls in this documentation. For more information on the VA FileMan DBS calls, see the *VA FileMan Developer's Guide*.

## 1.1 VA FileMan Reserved Symbols Used to Implement File Access Security

VA FileMan only recognizes the following two symbols as reserved symbols used to implement File Access Security:

- **At-Sign (@)**—**Programmer** access in VistA is defined as **DUZ(0)="@"**. It grants the privilege to become a programmer in VistA. **Programmer** access allows you to work outside many of the security controls enforced by VA FileMan, enables access to all VA FileMan files, access to modify data dictionaries, etc. It is important to proceed with caution when having access to the system in this way.
- **Caret (^)**—The caret (^) overrides the At-Sign (@). For example, if the user's **DUZ(0)="@"**, but **WRITE ("WR")** access is set to a caret (^), that user *cannot* WRITE (such as using the VA FileMan **Enter or Edit File Entries** [DIEDIT] option)

to that file. This is often used at the field level, obstructing all **DUZ(0)** access to that field.

All other symbols/characters are available to use to implement site File Access Security via Classic FileMan APIs at the developer or VA Facilities discretion. **DUZ(0)** is set after the user has been validated by Kernel. The value for any user's **DUZ(0)** can be found in the FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON (#200) file.



**REF:** For information about **DUZ(0) = "@" Programmer** access in VistA, see the *VA FileMan Developer's Guide*, on the VA Software Document Library (VDL): [FileMan \(DI\) Application](#).

## 2 What Type of File Access Security is Your Site Using?

Security control for file access at VA facilities is implemented via either of the following two methods:

- [Classic VA FileMan File Access Security](#)
- [Kernel File Access Security](#)

To know how to manage file security at your site, it is first necessary to determine which security implementation your site is using. An easy way to determine if the **Kernel File Access Security** [XUFILEACCESS] option has been implemented is to use the VA FileMan **Data Dictionary Utilities** [DI DDU] option, as shown in [Figure 1](#).

Select the **List File Attributes** [DILIST] option. At the "START WITH WHAT FILE:" prompt, select a VA FileMan file from which to display the data dictionary. If your site is using Kernel File Access Security, the following message appears after the file description and security access information is displayed, [Figure 1](#):

(NOTE: Kernel's File Access Security has been installed in this UCI.)



**NOTE:** The file used to create the screen capture in [Figure 1](#) is fictitious and used only for the purposes of this example.

**Figure 1: Example—Verify Site is Using Kernel File Access Security**

```
VA FileMan Version 22.2

Enter or Edit File Entries
Print File Entries
Search File Entries
Modify File Attributes
Inquire to File Entries
Utility Functions ...
Data Dictionary Utilities ...
Transfer Entries
Other Options ...

Select VA FileMan <TEST ACCOUNT> Option: DATA <Enter> Dictionary Utilities

List File Attributes
Map Pointer Relations
Check/Fix DD Structure
Find Pointers into a File
Update the META Data Dictionary

Select Data Dictionary Utilities <TEST ACCOUNT> Option: LIST <Enter> File
Attributes

START WITH WHAT FILE: FILE// EXAMPLE <Enter>
GO TO WHAT FILE: EXAMPLE// <Enter>
Select SUB-FILE: <Enter>
Select LISTING FORMAT: STANDARD// <Enter>
Start with field: FIRST// <Enter>
DEVICE: <Enter> Right Margin: 80// <Enter>
STANDARD DATA DICTIONARY #123 -- EXAMPLE FILE 1
STORED IN ^MYGLOBAL(123, (999 ENTRIES) SITE: <REDACTED>.VA.GOV UCI: NVS,NOU

DATA          NAME          GLOBAL          DATA
ELEMENT       TITLE           LOCATION        TYPE
-----
This is an example file.

          DD ACCESS: @
          RD ACCESS: D
          WR ACCESS: d
          DEL ACCESS: d
          LAYGO ACCESS: d
          AUDIT ACCESS: @

Security access information.

(NOTE: Kernel's File Access Security applies to this File.)

Message indicates this VA site has implemented Kernel File Access Security.

[Data Dictionary listing continues as it would normally...]
```

If your VA Facility has implemented Kernel File Access Security, it is *not* using Classic VA FileMan File Access Security.

**i** **NOTE:** Developers can also verify Kernel File Access Security by checking for the **^VA(200,"AFOF")** cross-reference. If it exists, then Kernel File Access Security has been implemented.

[Figure 2](#) shows user field values in the NEW PERSON (#200) file using the same (fictitious) EXAMPLE (#123) file from [Figure 1](#):

**Figure 2: Example—Security Property Values in the NEW PERSON (#200) File**

NEW PERSON LIST		JUN 10,2025 12:18			PAGE 1		
NAME	FILE MANAGER ACCESS CODE						
FILE	DD ACCESS	DEL ACCESS	LAYGO ACCESS	RD ACCESS	WR ACCESS	AUDIT ACCESS	
FMUSER,ONE				dS			
EXAMPLE				YES			

A **NULL** value is equal to "**NO**". In [Figure 2](#), the **DUZ(0)** for the user named **ONE FMUSER** is equal to "**dS**." Based on the security access shown in [Figure 1](#), this indicates that the user does not have **READ (RD)** access.

**i** **REF:** For information on the difference between Kernel and VA Classic FileMan File Access Security, see the "[Kernel File Access Security vs. Classic VA FileMan File Access Security](#)" section.

**i** **REF:** For information about Kernel File Access Security, see the "File Access Security" section in the [Kernel 8.0 Systems Management: Signon/Security User Guide](#).

## 3 How to Use the File Access Security at Your Site

How to use the File Access Security at your site:

- If your VA Facility is not using Kernel File Access Security, see the "[Classic VA FileMan File Access Security](#)" section.
- If your VA Facility is using Kernel File Access Security, see the "[Kernel File Access Security](#)" section.

### 3.1 Classic VA FileMan File Access Security

Given the example in [Figure 1](#), if you *do not* see the following message when displaying a VistA file in the VA FileMan **Data Dictionary Utilities** option, then your site has *not* implemented Kernel File Access Security:

(NOTE: Kernel's File Access Security has been installed in this UCI.)

This means that File Access Security at your site is implemented based on VA FileMan symbol(s) that are set into the security access property for a particular file via Classic VA FileMan APIs. User file access is based upon the VA FileMan symbols contained in the **DUZ(0)** value.

#### Example

In this example:

- The **READ** ("**RD**") access for the (fictitious) EXAMPLE (#123) file in Figure 1, is equal to "**D**,"
- The user's **DUZ(0)** is equal to "**AaZz**."

This means that the user is not able to view the data via VA FileMan's exported menus, because the **DUZ(0)** does not contain the symbol "**D**." If that same user's **DUZ(0)** was equal to **AaDZz**, then the user is authorized to view the file's data via the VA FileMan's exported menus, because the **DUZ(0)** contains the symbol "**D**."

### 3.1.1 Kernel File Access Security vs. Classic VA FileMan File Access Security


In the Classic VA FileMan File Access Security environment, the user (shown in the previous [example](#)) would *not* be authorized to view the data contained in the (fictitious) EXAMPLE (#123) file using the VA FileMan exported menu options, because the user's (**ONE FMUSER**) FILEMANAGER ACCESS CODE (#3) field in the NEW PERSON (#200) file does not contain the symbol "D." However, in a Kernel File Access Security environment, the file **READ** ("RD") access is set to **YES** for the user. Hence, the user would be able to view the data.


### 3.1.2 File Level Security Properties

[Table 2](#) lists the six File Access Security properties involved with File Access Security. If a File Access Security property is *not* defined, the value is **NULL**. In the Classic VA FileMan File Access Security environment, properties with **NULL** values allow full user access to the VA FileMan exported menu options for that property.

**Table 2: File Level Security Properties—Classic VA FileMan File Access Security**

Access	Security Property Description	Property Location (Classic VA FileMan)
<b>AUDIT</b>	<p>The <b>AUDIT</b> security property controls the setting of auditing characteristics and the deletion of audit trails. This property only deals with the auditing of data and <i>not</i> the auditing of data dictionary (DD) changes. To audit DD changes, users would enter "YES" at the "DD AUDIT? NO// " prompt when modifying a file's File Security Access. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Fields Being Audited</b> [DIAUDITED FIELDS]</li> <li>• <b>Data Dictionaries Being Audited</b> [DIAUDIT DD]</li> <li>• <b>Purge Data Audits</b> [DIAUDIT PURGE DATA]</li> <li>• <b>Purge DD Audits</b> [DIAUDIT PURGE DD]</li> </ul>	$\wedge\text{DIC}(\langle\text{file number}\rangle,0,\text{"AUDIT"})=\langle\text{value}\rangle$

Access	Security Property Description	Property Location (Classic VA FileMan)
	<ul style="list-style-type: none"> <li>• <b>Turn Data Audit On/Off</b> [DIAUDIT TURN ON/OFF]</li> </ul>	
<b>DATA DICTIONARY ("DD")</b>	<p>The DATA DICTIONARY security property controls who has access to modify the data dictionary. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Modify File Attributes</b> [DIMODIFY]</li> <li>• <b>Utility Functions</b> [DIUTILITY]</li> <li>• <b>Data Dictionary Utilities</b> [DI DDU]</li> </ul> <p>For example, to use the <b>Map Pointer Relations</b> option, <b>DD</b> access is needed to the PACKAGE (#9.4) file and to the files one selects for mapping.</p>	$\wedge$ DIC(<file number>,0,"DD")= <value>
<b>DELETE ("DEL")</b>	<p>The <b>DELETE</b> security property controls who can delete an existing record that is contained within the file. It does <i>not</i> permit deletion of the file or any of its attribute fields. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Enter or Edit File Entries</b> [DIEDIT]</li> <li>• <b>Transfer Entries</b> [DITRANSFER]</li> </ul>	$\wedge$ DIC(<file number>,0,"DEL")= <value>
<b>LAYGO</b>	<p>The <b>LAYGO</b> (Learn As You Go) security property controls who can add a new record to the file. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• Enter or Edit File Entries [DIEDIT]</li> </ul> <p> <b>NOTE:</b> You <i>must</i> have <b>LAYGO</b> and <b>WRITE</b> access to a file to add new entries. In addition, you <i>must</i> have <b>WRITE</b> access at the field level for all required identifier fields.</p>	$\wedge$ DIC(<file number>,0,"LAYGO")= <value>

Access	Security Property Description	Property Location (Classic VA FileMan)
<b>READ ("RD")</b>	<p>The <b>READ</b> security property controls who has access to read data contained within a file. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Print File Entries</b> [DIPRINT]</li> <li>• <b>Search File Entries</b> [DISEARCH]</li> <li>• <b>Inquire to File Entries</b> [DIINQUIRE]</li> <li>• <b>Statistics</b> [DISTATISTICS]</li> <li>• <b>List File Attributes</b> [DILIST]</li> <li>• <b>Transfer Entries</b> [DITRANSFER]</li> </ul> <p>To transfer text, the user needs <b>READ</b> access to the file from which text is being transferred. Similarly, <b>WRITE</b> access is needed for the file to which entries are being transferred with this option.</p> <p>Transfer File Entries (transfer-to file)</p> <p> <b>NOTE:</b> <b>READ</b> access is also required to use some of the Filegram and Audit options.</p>	$\wedge$ DIC(<file number>,0,"RD")= <value>
<b>WRITE ("WR")</b>	<p>The <b>WRITE</b> security property controls who can alter data in an existing record that is contained within the file. It will not permit the adding of new entries to the file. Examples of the VA FileMan options that this property controls are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Enter or Edit File Entries</b> [DIEDIT]</li> <li>• <b>Transfer Entries</b> [DITRANSFER]</li> </ul> <p>To transfer text, the user needs <b>READ</b> access to the file from which text is being transferred. Similarly, <b>WRITE</b> access is needed for the file to which</p>	$\wedge$ DIC(<file number>,0,"WR")= <value>

Access	Security Property Description	Property Location (Classic VA FileMan)
	entries are being transferred with this option. Transfer File Entries (transfer-to file) Compare/Merge File Entries	

## 3.2 Kernel File Access Security

Given the example in [Figure 1](#), if you *do* see the following message when displaying a VistA file in the VA FileMan Data Dictionary Utilities option:

(NOTE: Kernel's File Access Security has been installed in this UCI.)

File Access Security is controlled by the ACCESSIBLE FILE (#32) Multiple in the NEW PERSON (#200) file. This means that File Access Security to a particular file is not based on the VA FileMan Access Code **DUZ(0)** value. Rather, a lookup is done on the user's ACCESSIBLE FILE (#32) Multiple record in the NEW PERSON (#200) file to determine which accesses are allowed to the file in question via VA FileMan exported menus. If the user's VA FileMan Access Code [that is **DUZ(0)**] is equal to the At-Sign (@), they are allowed access to all files.



**NOTE:** Kernel File Access Security is known as an Access Control List in other systems.



**REF:** For more information on Kernel File Access Security, see the "File Access Security" section in the [Kernel 8.0 Systems Management: Signon/Security User Guide](#).

### 3.2.1 File Level Security Properties

[Table 2](#) lists the six security properties involved with File Access Security for Kernel that are equivalent to the Classic VA FileMan File Access Security. Unlike Classic VA FileMan File Access Security, in a Kernel File Access Security environment, if the security property is not defined (that is the value is **NULL**), the VA FileMan exported menu options for that property are not open to full access for users.



**NOTE:** These same security properties left undefined in a Classic VA FileMan File Access Security environment allow the related VA FileMan exported menu options up to full access for users.