Kernel 8.0 and Kernel Toolkit 7.3

Systems Management Guide

January 2024

Department of Veterans Affairs (VA)

Office of Information and Technology (OIT)

Software Product Management (SPM)

Revision History

Date	Revision	Description	Author
01/23/2024	7.16	Updates for Kernel Patch XU*8.0*756: - Section 3.6: Added enhancements made with Kernel Patch XU*8.0*756. - Figure 12: Added SIGN-ON LOG RETENTION field.	VistA Infrastructure Shared Services (VISS) Development Team
09/21/2023	7.15	Updates: - Added Section 10.2.2, "Troubleshooting Using ALERT TRACKING (#8992.1) File." - Added Section 10.2.2.1, "Simple Alert to INITIAL RECIPIENT." - Added Section 10.2.2.2, "Alert to Surrogate—Who Processes the Alert." - Added Section 10.2.2.3, "Alert to Surrogate—Who Ignores Alert."	VistA Infrastructure Shared Services (VISS) Development Team
06/01/2023	7.14	Updates: Removed all prior content updates for Kernel Patch XU*8.0*663 (03/10/2023), since that patch release was changed to "Released In Error."	VISS Development Team
03/14/2023	7.13	Updates for Kernel Patch XU*8.0*772: - Section 10.2.1.2: Removed reference to purging the ALERT TRACKING (#8992.1) file. - Updated document dates.	VISS Development Team
03/10/2023	7.12	Updates: - Modified Section 3.2.1, "Add a New User to the System Option." - Modified Section 3.2.2, "Grant Access by Profile Option." - Modified Section 3.3, "Edit an Existing User Option." - Updated references to Kernel and Kernel Toolkit manual file names: from "Kernel 8.0 & Kernel Toolkit 7.3 … " to "Kernel 8.0 and Kernel Toolkit 7.3 … " for all occurrences.	- Master Veteran Index (MVI) Development Team - VISS Development Team

Date	Revision	Description	Author
10/03/2022	7.11	Updates for Kernel Patch XU*8.0*775: - Section 3.2.2, "Grant Access By Profile Option:" Added note about the exclusion of the PSDRPH security key allocation. - Section 6.4.1, Figure 61: Updated the out of order options in the **XU EPCS UTILITY** menu and options out of order note. - Section 8.12.3, "Allocating/De-Allocating Security Keys:" Added two notes about the exclusion of the PSDRPH security key allocation. - Section 9.1.5, "Copy One Users Menus and Keys to Others:" Added note about the exclusion of the PSDRPH security key allocation. - Updated organizational references (e.g., changed "Enterprise Program Management Office [EPMO]" to "Software Product Management (SPM)") throughout.	- Liberty IT Solutions - VISS Development Team
09/01/2021	7.10	Updates related to Kernel Patch XU*8.0*753: Section 2.6, "Edit User Characteristics Option": - Figure 8: Edit User Characteristics Option—ScreenMan Form: Added the DISABILITY USER display field to the ScreenMan form. - Table 3: Edit User Characteristics Option—Editable Fields: Added the DISABILITY USER (#508.1). Updates related to Kernel Patch XU*8.0*738: - Section 24.7.8, "Backing Up Transport Globals:" Modified intro text. - Added Figure 321. Updates related to Kernel Patch XU*8.0*693:	VistA Infrastructure (VI)/VistA Kernel Development Team

Date	Revision	Description	Author
		- Section 3.4.2.1: Added references to the bulletins notifications. - Added Figure 33 and Figure 34. Updates related to Kernel Patch XU*8.0*707: - Section 4.3.3; Step 3: File range changed from File #2 to File #1.1 equal or greater. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
07/16/2020	7.09	Updates related to Kernel Patch XU*8.0*730: - Updated Section 10.1.4. - Added Section 10.1.4.1. - Added Section 10.1.4.2.	VI/VistA Kernel Development Team
02/20/2020	7.08	- Updates related to Kernel Patch XU*8.0*701: ○ Restructured Section 3.6.2. Added new Sections 3.6.2.1 and 3.6.2.2. ○ Section 3.6.2.2: Added Caution Note related to PIV authentication regarding the frequency of running the XUFPURGE option. ○ Moved Sections 3.6.2.2.1 and 3.6.2.2.2 under Section 3.6.2.2. - Updates related to the release of Kernel Patches XU*8.0*607 and 608: ○ Section 14.2.2.2: Renamed section and added reference and example to setting the MULTIPLE SIGN-ON field for the Lock Manager. ○ Created new Section 14.8, "Troubleshooting," and the first issue, Section 14.8.1, "Node Connection Error." **Software Versions:**	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		Kernel 8.0 **Toolkit 7.3**	
02/06/2020	7.07	Tech Edits for Kernel Patches XU*8.0*701: • Updated Figure 12. • Updated Section 3.1.2.16; added references to the STRICT TOKEN VALIDATION (#220) field and **cache.cer** file. • Updated Table 6; added the CREDENTIAL TYPE (#102) and CREDENTIAL WARNINGS (#103) fields. • Updated Figure 36. • Updated Section 3.6.2: Added new subsections: 3.6.2.2.1, 3.6.2.2.2, and 3.6.2.3. • Updated Figure 38. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
12/04/2019	7.06	Tech Edits for Kernel Patches XU*8.0*607 and 608: Kernel Lock Manager Utility. Added the following: • Section 14, "Lock Manager Utility" and all sub-sections. • Updated all TOCs and cross-references. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
10/11/2018	7.05	Tech Edits for Patch XU*8.0*690: • Updated Section 10.1.1, "Processing Alerts." • Added Section 10.1.1.1,"Critical Alerts." • Section 10.2.1.5.1, "Critical Alerts Count Report Option:" Added reference to the ALERT CRITICAL TEXT (#8992.3) file.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		- Section 10.2.1.5.2, ”List Alerts for a user from a specified date Option:” Updated the description. - Section 10.2.1.5.4, “User Alerts Count Report Option:” Added reference to the ALERT CRITICAL TEXT (#8992.3) file. - Section 10.2.1.5.1.1, “Error Handling—Missing SERVICE/SECTION Data:” Added error handling of missing SERVICE/SECTION (#29) field data. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
08/22/2018	7.04	Updates for Kernel Patch XU*8.0*679: - Added Section 5.1.1.1, “Electronic Signature Code Edit Restrictions:” to Describes restrictions to the SIGNATURE BLOCK PRINTED NAME (#20.2) and SIGNATURE BLOCK TITLE (#20.3) fields. - Added Section 5.2.1.1, “Electronic Signature Block Edit Restrictions:” Describes restrictions to the DEGREE (#10.6), SIGNATURE BLOCK PRINTED NAME (#20.2), and SIGNATURE BLOCK TITLE (#20.3) fields. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
03/01/2018	7.03	Tech Edits: - Updated Section 2, “Signon/Security: User Interface” and Section 3.4.2, “Automatically Deactivating Users,” with regard to “smart card (aka PIV card) signons. - Updated Section 2.1.1, “Defining a Strong Verify Code:” Includes references to other section regarding Verify code expiration and option to reset.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		- Updated Section 3.1, "Signon Process." - Updated Figure 36 and added Table 6 in Section 3.5.4, "Print Sign-on Log Option," based on updates made with Kernel Patch XU*8.0*630. - Updated Section 1; Kernel (and soon all of VistA) is no longer vendor-independent. Much of the new work being done in Kernel and other namespaces relies on Cache ObjectScript. - Added the "Parameter Tools" section taking content from the *Parameter Tools Supplement to Patch Description: Patch XT*7.3*26* document (ktk7_3p26sp.pdf). - Updated Sections 1.1, 2, 2.1, 2.1.1, 3.1.2.16, and 3.5.8 to add or clarify references to 2-Factor Authentication (2FA) vs. use of the Access and Verify codes. - Updated styles and formatting throughout. - Updated all TOCs, lists, cross-references, etc. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
08/10/2016	7.02	Tech Edits: - Updated Section 16.1.3.2 and 16.6.4.2 for additional HOST file examples and clarifications. - Updated all TOCs, lists, cross-references, etc. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
08/09/2016	7.01	Tech Edits based on Kernel patches XU*8.0*655, 659, and 667: - Updated Directive 6402 reference in the "Software Disclaimer" section.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		- Updated/Added the following sections for 2-Factor Authentication (2FA)-related information: 2.1, 3.1, 3.1.2.16 (new), 3.5.4, 4.2.8.1, 6.7.1, 6.7.2, and 6.7.3. - Updated Section 2.1.1 to indicate that the caret (**^**) is a reserved symbol and added a reference to and VA Directive and Handbook 6500. - Updated Section 2.1.1.1 for long password future changes pending. - Updated Figure 8. - Table 3: Added TITLE and ELECTRONIC SIGNATURE COD fields. - Updated Figure 13 for 2-Factor Authentication (2FA). - Table 4: Added the NETWORK USERNAME field. - Updated Figure 28. - Section 3.4.2.2: Added Cautionary note. - Sections 3.5.8 and 3.5.11: Added references to Broker Security Enhancement (BSE). - Section 6.1: Updated references to CPRS documentation. - Section 6.4.15: Updated the "XU USER START-UP Option" section; merged from (deleted). **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
07/19/2016	7.0	Updates: - Moving the "System Management Menus" section and sub-sections from this document into the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual*. - Updated Figure 57 and Figure 59: Removed extraneous/test-only options from the menu.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		• Updated the “Prohibited Times” section: Included information regarding TaskMan scheduled options. • “Software Disclaimer” section: Added Caution note regarding modification of Kernel routines. • Converted Word document to .docx format. • Reformatted document to follow latest documentation standards and formatting rules. Also, formatted document for online presentation vs. print presentation (i.e., for double-sided printing). These changes include: o Revised section page setup. o Removed section headers. o Revised document footers. o Removed blank pages between sections. o Revised all heading style formatting. • Updated organizational references (e.g., “Product Development [PD]” to “Enterprise Program Management Office [EPMO]). • Redacted document for the following information: o Names (replaced with role and initials). o Production IP addresses and ports. o VA Intranet websites. o Server geographic locations and node names. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
05/31/2013	6.2	Updates:	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		• Updates for patch XU*8.0*614 based on feedback from developer: ○ Added the **Single User Menu Tree Rebuild** [XQBUILDUSER] option to the “Single User Menu Tree Rebuild” It was attached to the Menu Rebuild Menu [XQBUILDMAIN] option. ○ Added the **Menu Rebuild Menu** [XQBUILDMAIN] option. ○ Added the **List Unreferenced Menu Options** [XQ LIST UNREFERENCED OPTIONS] option. ○ “Menu Startup Parameter” section: Added the **XQ MENUMANAGER PROMPT** parameter. • Added the “System Management Menus” section. This section lists and briefly describes all Kernel operations, management, user, and developer-related menus and options. It also includes cross-reference links to existing sections that further describe a menu or option elsewhere in this document. • Renamed and updated the “User Management Menu” section. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
04/30/2013	6.1	Tech edit updates: • XU*8.0*580: Updated document for Kernel patch XU*8.0*580 in support of the Drug Enforcement Agency (DEA) e-Prescribing of Controlled Substances (CS) (ePCS) using Public Key Infrastructure (PKI). ○ Added the “DEA ePCS Utility” section with the following subsections: – Overview – Processes	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		– Configuring the DEA ePCS Utility, including instructions to: ▪ Set the XUEPCS REPORT DEVICE Parameter. ▪ Add DEA ePCS Utility Users. – Using the DEA ePCS Utility (includes description of all Menus/Options). – Prescription Validation and Verification Process—PKIServer.exe Application – PIV Card Validation—Revocation Server – Windows Authentication and Cryptographic Operations • Reformatted document to follow current style guide and standards. • Replaced references from "*VA FileMan Getting Started Manual*" to "*VA FileMan User Manual*," since the next VA FileMan 22.*n* software version will create a new "*VA FileMan Getting Started Manual*." • Patches XU*8.0*602: Updated the following sections, as per developer: ○ "Processing Alerts" section. ○ "Surrogates and Alerts" section. • Updated the "Understanding DUZ (User Number)" section to give a more detailed explanation and examples of the **DUZ** array. • Updated the "KEEP AT TERMINATE" section as per email. • Patch XU*8.0*546: Support for Device Hunt Groups was removed. This includes removal of the *HUNT GROUP (#29) and HUNT GROUP DEVICE (#30) fields in the DEVICE (#3.5) file. Sites had to remove any HUNT GROUP devices before installing this patch using VA FileMan to find any existing Hunt	

Date	Revision	Description	Author
		Groups. “Hunt Groups” section was deleted from this manual. Also, any references to “Hunt Groups” were removed. - Added blue font highlighting and underline to signify internal links to figures, tables, or sections for ease of use, similar to what one sees to hyperlinks on a Web page. - Updated document for Section 508 conformance using word’s built-in Accessibility check: ○ Added table bookmarks. ○ Added screen tips for all URL links. ○ Changed all floating callout boxes to in-line, causing reformatting of numerous dialog screen captures. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
06/06/2012	6.0	Updates: - Added the “XU USER START-UP Option” section. The XU USER START-UP option was added with Kernel patch XU*8.0*593. - Added Section 16.6.4 “Verify HFS and NULL Device Setup *(required)*,” in the “Troubleshooting” section in “Device Handler: System Management.” - Updated all VA organizational references. - Revised all version numbers in the “Revision History” section. - Updated the “Orientation” section. - Updated the overall document for current national documentation standards and style guides. For example: ○ Changed all Heading *n* styles to use Arial font.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		○ Changed all Heading *n* styles to be left justified. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
03/22/2010	5.3	Updates: • Added Section 25.3, "Edit Install Status Option released with Kernel patch XU*8.0*539. • Added Figure 330: Edit Install Status Option—Sample User Dialog. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
11/16/2009	5.2	Updates: • Updated references to the CHCKSUM^XTSUMBLD direct mode utility throughout. • Updated organizational references. • Minor format updates (e.g., reordered document revision history table to display latest to earliest, added outline numbering). • Other minor format updates to correspond with the latest standards and style guides. • Updated the "Automatically Deactivating Users" section in "Signon/Security: System Management" for Kernel patch XU*8.0*514. • Re-ordered and edited all topics in the "Device Handler: System Management" section. Also, added updates to the Device handler based on Kernel patch XU*8.0*440. • Moved the following section content from the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* to the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*, because the functions documented	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		are more developer-related than system management-related: ○ Miscellaneous Programmer Tools: Programmer Options Menu and %Z Editor; see the “Miscellaneous Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*. ○ Routine Tools; see the “Routine Tools” section in the “Toolkit: Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*. ○ Verification Tools; see the “Verification Tools” section in the “Toolkit: Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*. ○ XGF Function Library; see the “XGF Function Library” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*. • Updated Section 9.1.3 and 9.1.6.2 for Kernel patch XU*8.0*482. • Reviewed and updated all sections for minor format changes (e.g., bulleted lists and tables), style updates, spelling, and grammar fixes. **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
06/10/2008	5.1	Updates: • Updated the “Defining a Strong Verify Code” section. • Updated the “File Access Security” section based on the newly created *VA FileMan (Version 22) and Kernel (Version 8.0) File Access Security* supplemental document on the VA Software Document Library (VDL). • Deleted “Default Task Priority” section from this manual.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		• Moved the "Error Screens" section from the "TaskMan: System Management—Operation" section to the "Error Processing" section. • Updated the "Alpha/Beta Tracking" section in Section 24. Merged information from the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* (this manual) into the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide* in order to avoid duplication and confusion with instructions/procedures. • Updated VA OIT organization changes and the document properties (e.g., Title, Author, Creation Dates, Keywords, etc.). • Updated references to the VDL. • Removed all references to HSD&D. • Updated Alert options in Figure 132 and added the missing descriptions for those Alert-related options. • Completed updates to remove obsolete references to MSM, PDP, 486, VAX Alpha, etc. and updated references to DSM for OpenVMS to Caché where appropriate. • Updated content references to checksum compares based on Kernel patch XU*8.0*393. • Changed references from "%INDEX" to "XINDEX" where appropriate. • Updated Section III, Device Handler. • Deleted "Kermit" section. • Updated "Special Queueing" section in "TaskMan: System Management—Operation" Added Table 45. • Updated "Security Forms" section in "Signon/Security: System Management." **Software Versions:**	

Date	Revision	Description	Author
		Kernel 8.0 **Toolkit 7.3**	
02/08/2007	5.0	The Kernel Toolkit documentation set is being combined with the Kernel documentation set. All Kernel Toolkit content will be moved to the appropriate Kernel manual section. In the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide*, the following Kernel Toolkit sections have been added to the Section VI, "Toolkit:" • Multi-Term Look-Up (MTLU) • Routine Tools • Verification Tools • Also Changed Kernel document title references to: ○ Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide (previously known as the Kernel Programmer Manual). ○ Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide (previously known as the Kernel Systems Manual). **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	VI/VistA Kernel Development Team
07/13/2006	4.0	Updates: • Made minor formatting updates throughout. • Changed the original "Other Tools" section to become the "Toolkit" section, see note below. • Added "Multi-Term Look-Up (MTLU)" and "Tools" sections from the original *Toolkit User Manual* (7.3), see note below. • Removed the "Response Time Measures" section from the original "Capacity Management" section in the *Toolkit User Manual* (7.3), see note below. Kernel Toolkit patch XT*7.3*102 removed all **Response**	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		Time Log Option menu options [**XURTL***]. - All Kernel Toolkit content currently in the Kernel Toolkit User Manual and Kernel Toolkit Technical Manual is being absorbed by the Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide, Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide, and Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual. Other Toolkit content has been replaced by other manual sets, including: o Duplicate Record Merge: Patient Merge o Resource Usage Monitor (RUM) o Statistical Analysis of Global Growth (SAGG) o Capacity Management (CM) Tools **Software Versions:** **Kernel 8.0** **Toolkit 7.3**	
02/03/2006	3.0	Updates: - Reformatted document to the latest SOP and Style Guidelines. - Updated files, routines, options, APIs, security keys, etc. **Software Version: Kernel 8.0**	VI/VistA Kernel Development Team
12/20/2004	2.1	Reviewed document and edited for the "Data Scrubbing" and the "PDF 508 Compliance" projects. **Data Scrubbing—**Changed all patient/user TEST data to conform to OIT standards and conventions as indicated below: The first three digits (prefix) of any Social Security Numbers (SSN) start with "000" or "666." Patient or user names are formatted as follows: XUPATIENT,[N] or XUUSER,[N] respectively, where the N is a number written out and incremented with each new entry	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		(e.g., XUPATIENT, ONE, XUPATIENT, TWO, etc.). Other personal demographic-related data (e.g., addresses, phones, IP addresses, etc.) were also changed to be generic. **PDF 508 Compliance—**The final PDF document was recreated and now supports the minimum requirements to be 508 compliant (i.e., accessibility tags, language selection, alternate text for all images/icons, fully functional Web links, successfully passed Adobe Acrobat Quick Check). **Software Version: Kernel 8.0**	
12/09/2004	2.0	Kernel 8.0 documentation reformatting/revision. This is the initial complete reformatting of the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* since its original release in July 1995. The largest change with the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* is that all developer-specific content has been extracted and placed into a new *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Also, at this point in time, only minimal content updates have been made based on select released Kernel patches. Due to time constraints, *not* all released Kernel patches with content changes have been added at this time. We wanted to get a new baseline document published so that in the future we can more easily update the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide*. As time allows, we will be updating this reformatted manual with all released patch information that affects its content. Because of the chapter-numbering scheme, future additions can be made with minimal disruption to the entire manual page flow.	VI/VistA Kernel Development Team

Date	Revision	Description	Author
		Thanks for your patience! **Software Version: Kernel 8.0**	
07/--/1995	1.0	Initial Kernel 8.0 software and documentation release **Software Version: Kernel 8.0**	VI/VistA Kernel Development Team

Patch Revisions

For the current patch history related to this software, see the Patch Module on FORUM.

Table of Contents

List of Figures

List of Tables

Orientation

How to Use this Manual

Throughout this manual, advice and instruction are offered about the numerous Kernel 8.0 and Kernel Toolkit 7.3 tools and functionality provided for the Veterans Health Information Systems and Technology Architecture (VistA) system management and end-users (e.g., site parameters).

The *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* is divided into six major sections, based on the following functional divisions within Kernel/Kernel Toolkit:

I. Signon/Security (e.g., techniques for granting user access and monitoring computing activity)

II. Menu Manager (e.g., techniques for managing menus)

III. Device Handler

IV. TaskMan

V. Kernel Installation and Distribution System

VI. Toolkit

REF: For information on developer tools (e.g., Direct Mode Utilities and Application Program Interfaces [APIs]), see the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet website.

Information on recommended system configuration and setting Kernel's site parameters, as well as lists of files, routines, options, and other components are documented in the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual*.

Information about managing computer security, which includes a detailed description of techniques that can be used to monitor and audit computing activity, is presented in the *Kernel Security Tools Manual*.

Instructions for installing Kernel are provided in the *Kernel Installation Guide*. This guide also includes information about software application management (e.g., *recommended* settings for site parameters and scheduling time frames for tasked options).

This manual is further organized within each section of Kernel in the following order:

1. User Interface—Information of relevance to general end-users.
2. System Management—Information of relevance to system managers.

When a subject is large enough (e.g., Signon/Security), separate sections are devoted to the “User Interface” and “System Management” topics. In other cases, where the subject matter is smaller (e.g., the discussion of the Browser device), the two divisions of audience are contained entirely within a section or sub-section.

Intended Audience

The intended audience of this manual is the following stakeholders:

- Software Product Management (SPM)—VistA legacy development teams.
- 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.
- Information Security Officers (ISOs)—Personnel at VA sites responsible for system security.
- Product Support (PS)—Personnel who support Kernel-related products.

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 VA Directive 6402 (pending signature), these routines 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 Field Office (OIFO).

Documentation Disclaimer

This manual provides an overall explanation of using Kernel; however, no attempt is made to explain how the overall VistA programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA Internet and Intranet SharePoint sites and websites for a general orientation to VistA. For example, visit the Office of Information and Technology (OIT) Software Product Management (SPM) Intranet Website.

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

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
ℹ	**NOTE / REF:** Used to inform the reader of general information including references to additional reading material.
⚠	**CAUTION / RECOMMENDATION / DISCLAIMER:** 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) 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 Kernel (XU or KRN) test patient and user names would be documented as follows:

KRNPATIENT,ONE; KRNPATIENT,TWO; KRNPATIENT,THREE; … KRNPATIENT,14; etc.

KRNUSER,ONE; KRNUSER,TWO; KRNUSER,THREE; … KRNUSER,14; etc.

- “Snapshots” of computer commands and online displays (i.e., 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 are **boldface** and (optionally) highlighted in yellow (e.g., **<Enter>**).
  - Emphasis within a dialog box is **boldface** and (optionally) highlighted in blue (e.g., STANDARD LISTENER: RUNNING).
  - Some software code reserved/key words are **boldface** 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 is 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 (e.g., XUPROGMODE security key).

  NOTE: Other software code (e.g., Delphi/Pascal and Java) variable names and file/folder names can be written in lower or mixed case (i.e., CamelCase).

Documentation Navigation

This document uses Microsoft® Word's built-in navigation for internal hyperlinks. To add **Back** and **Forward** navigation buttons to your toolbar, do the following:

1. Right-click anywhere on the customizable Toolbar in Word 2007 or higher (*not* the Ribbon section).
2. Select **Customize Quick Access Toolbar** from the secondary menu.
3. Select the drop-down arrow in the "Choose commands from:" box.
4. Select **All Commands** from the displayed list.
5. Scroll through the command list in the left column until you see the **Back** command (circle with arrow pointing left).
6. Select/Highlight the **Back** command and select **Add** to add it to your customized toolbar.
7. Scroll through the command list in the left column until you see the **Forward** command (circle with arrow pointing right).
8. Select/Highlight the **Forward** command and select **Add** to add it to the customized toolbar.
9. Select **OK**.

You can now use these **Back** and **Forward** command buttons in the Toolbar to navigate back and forth in the Word document when selecting hyperlinks within the 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 through the use of Kernel, MailMan, and VA FileMan utilities.

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

REF: See the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual* for further information.

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” section in the “File Management” section in 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 Kernel should consult the following:

- *Kernel Release Notes*
- *Kernel Installation Guide*
- *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* (this manual)
- *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*
- Kernel VA Intranet Website.

  This site contains other information and provides links to additional documentation.

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: Adobe Website

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

REF: Kernel manuals are located on the VDL at: VDL Kernel Application Documents

VistA documentation and software can also be downloaded from the Product Support (PS) Anonymous Directories.

1 Introduction

This manual provides descriptive information about Kernel 8.0 and Kernel Toolkit 7.3 for use by system administrators, application developers, Automated Data Processing Application Coordinators (ADPACs), and other end-users.

This manual assumes that the reader is familiar with the computing environment of the VA's Veterans Health Information Systems and Technology Architecture (VistA) and understands VA FileMan data structures and terminology. Some understanding of the M programming language is helpful for some parts of the manual. No attempt is made to explain how the overall VistA programming system is integrated and maintained; such methods and procedures are documented elsewhere. This manual does, however, provide an explanation of Kernel utilities, describing how they can be used to establish a standard user interface, monitor and manage the computer system, customize the environment according to local site needs, and define new areas of computing activities for users.

Kernel is an applications development environment, as well as a run-time environment providing standard services to applications software. It is *not* an operating system, but a set of utilities and associated files that are executed in an M environment. Kernel is central to VA VistA software strategy; in that it permits any VistA software application to run without modification on any hardware/software platform that supports American National Standards Institute (ANSI) Standard M. All operating system-specific or hardware-specific code is isolated to Kernel. Therefore, porting VistA to a new environment requires modification only to a handful of Kernel routines.

As a whole, Kernel provides a computing environment that permits controlled user access, presents menus for choosing from various computing activities, allows device selection for output, enables the tasking of background processes, and offers numerous tools for system management and application programming. Kernel also provides tools for software distribution and installation.

VistA users see the same user interface, regardless of the underlying system architecture, because VistA applications are built using Kernel facilities for signon, database access, option selection, and device selection. As a result, user interaction with the system is constant across VistA applications.

1.1 Users

Kernel provides the doorway into the VistA computer system, the menus that tie together the options and utilities to enhance those options.

For the doorway, Kernel provides the 2-Factor Authentication (2FA) system that you use to establish your identity to the VistA computer system.

Once you have signed on, Kernel provides your menus. Each user on the computer system, as identified by their Microsoft® Windows Active Directory profile, has their own individual set of menus and options.

The person or department managing the computer system organizes each user's menus. From your menu, you can run any application the computer system managers have made available to you. Kernel’s menu system is what is used to make VistA applications (e.g., Scheduling, Nursing, and Personnel) available to users.

To produce output from VistA applications (e.g., to printers or to the terminal screen), Kernel provides a common device interface called the Device Handler. To queue a job rather than run it directly, the Device Handler links to a common queuing system called TaskMan.

This manual contains information about these and other parts of Kernel. The intent of this manual is to help you learn to use Kernel and take fullest advantage of the facilities it provides. This manual also includes information for system managers and developers; to find the information of interest to you, the general user, look for sections and sub-sections containing the phrase “User Interface” in their titles.

ADP Application Coordinators (ADPACs) may want to skim through the *Kernel 8.0 and Kernel Toolkit 7.3 Systems Management Guide* and concentrate on the user interface sections and sub-sections, particularly issues concerning every Kernel user (e.g., signon process and menu navigation).

1.2 System Managers

Kernel provides the backbone of an M computing platform, providing a mechanism to organize M programs as options, and a way to organize those options into a menu system for users. Kernel provides the following major system management components:

- Alerts provide an integrated notification system.
- Device Handler provides a common device interface.
- Electronic Signature Codes provide a secure electronic approval system.
- File Access Security system manages access to VA FileMan files.
- Kernel Installation and Distribution System (KIDS) provides an application distribution and installation system.
- Menu Manager provides a common menu management system.
- Signon/Security organizes users and allows secure logons.
- TaskMan provides a common job queuing system.

Kernel provides the system manager the means to manage a secure, multi-user M-based computer system. Some typical daily tasks performed by system managers using Kernel system management tools include:

- Setting up accounts for new users and terminating accounts for expired users.
- Adding and subtracting options from users' menus.
- Controlling file access for users.
- Monitoring TaskMan task queues.
- Terminating unwanted tasks.
- Monitoring devices.
- Creating and modifying links to output devices in the DEVICE (#3.5) file.
- Installing software applications.

Within sections and sub-sections of this manual you can find general user information in the "User Interface" section and system manager information in the "System Management" section.

REF: For information on developer tools (e.g., Direct Mode Utilities and Application Program Interfaces [APIs]), see the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet Website.

Information on recommended system configuration and setting Kernel's site parameters, as well as lists of files, routines, options, and other components are documented in the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual*.

Information about managing computer security, which includes a detailed description of techniques that can be used to monitor and audit computing activity, is presented in the *Kernel Security Tools Manual*.

Instructions for installing Kernel are provided in the *Kernel Installation Guide*. This guide also includes information about software application management (e.g., recommended settings for site parameters and scheduling time frames for tasked options).

I. Signon/Security

2 Signon/Security: User Interface

The first step you take each time you access the computer system is called signing on. When you sign on to the VistA computer system, you are required to use the appropriate user credentials:

- Access and Verify codes
- 2-Factor Authentication (2FA)—Digital certificate in a VA-approved smart card, such as the Personal Identification Verification (PIV) smart card plus a Personal Identification Number (PIN).

NOTE: Access and Verify codes is the fallback signon in cases when 2FA signon is *not* available.

These credentials identify you to the computer system, and, as these credentials are private to you, serve to prevent unauthorized access to your account.

NOTE: Because Access and Verify code authentication is less secure than 2FA, their use may be deprecated and disabled at some future date.

You are shielded from most steps in the signon process. In the background, Kernel's Signon/Security does the following:

- Establishes the proper environment.
- Records and monitors the signon event.
- Takes you to Menu Manager, which presents a list of menu options that let you interact with other parts of Kernel and software applications.

When you complete a session on the computer system, you sign out to exit.

2.1 Signing On

To authenticate yourself to VistA (Kernel's "front door"), you need to sign onto the system. The user signon (authentication) interface varies based on the type of Vista application software being run:

- 2-Factor Authentication (2FA)—VistA supports delegated 2-Factor Authentication (2FA) through Identity and Access Management (IAM). A smart card containing Public Key Infrastructure (PKI) digital certificates combined with a private security is used to authenticate and uniquely identify the user. The user is prompted for a Personal Identification Number (PIN) to unlock the security key and authenticate. This method of authentication provides a higher level of security and takes precedence over all other forms of authentication. As client applications are migrated to 2-Factor Authentication (2FA), other forms of authentication may be deprecated and disabled.
- Character User Interface (CHUI)-based applications—This includes M-based roll-and-scroll applications used to access Kernel on the VistA M Server (e.g., Laboratory, Pharmacy). With this type of authentication interface, users are first prompted with an "ACCESS CODE:" prompt. Entering an Access code and pressing the **<Enter>** key brings up the "VERIFY CODE:" prompt.

  REF: For a sample of the roll-and-scroll signon prompts, please see Figure 1.

- Graphical User GUI client/server applications—This includes rich client or client/server applications used to access Kernel on the VistA M Server via RPC Broker (Delphi/Pascal)- or VistALink (Java)-based components (e.g., Computerized Patient Record System [CPRS] or Care Management). With this type of authentication interface, users are presented with a GUI signon dialog box. Users can click in or tab to the Access and Verify code entry fields and press **OK**.

  REF: For a sample of the RPC Broker signon dialog box and more information on RPC Broker, see the RPC Broker documentation located on the VA Software Document Library (VDL) at: VDL RPC Broker Application Documents

- Web-based applications—This includes Web-based applications that use a client Web browser and Kernel Authentication and Authorization Java (2) Enterprise Edition (KAAJEE) to access Kernel on the VistA M Server (e.g., Blind Rehab). With this type of authentication interface, users are presented with a GUI signon dialog Web page. Users can click in or tab to the Access and Verify code entry fields and press **Login**.

  REF: For a sample of the KAAJEE signon dialog Web page and more information on KAAJEE, see the KAAJEE documentation located on the VA Software Document Library (VDL) at: VDL KAAJEE Application Documents

Figure 1 shows a sample of the roll-and-scroll signon prompts. Your Access code establishes your unique identity to Kernel. Your matching Verify code corroborates your identity completing the VistA Kernel authentication process. Asterisks only are displayed when you enter your Access and Verify codes, so that the actual characters are *not* displayed (echoed back) on the screen. Codes are encrypted after they are entered and compared with the encrypted stored values for a match.

REF: For a description of valid and strong Verify code, see the "Defining a Strong Verify Code" section.

Figure 1: Signing on to VistA—Sample Roll-and-Scroll User Authentication Dialog

```
ACCESS CODE: ********
VERIFY CODE: ********
Device: _TNA8628: <Enter>
Not a valid ACCESS CODE/VERIFY CODE pair.
```

An invalid Access and Verify code pair produces an error.

```
ACCESS CODES: ********
VERIFY CODES: ********
Good evening FRIEND     You last signed on Apr 21,1992 at 07:57

There was 1 unsuccessful attempt since you last signed on:

You were last executing the 'MailMan Menu' menu option.
Do you wish to resume? YES//
```

Entering a valid Access and Verify code combination completes the signon authentication process and takes you beyond Signon/Security into Kernel's Menu Manager (or other security role-based access keys) used to authorize your appropriate level of access to data or application functionality.

If you have *not* been assigned a primary menu, Kernel displays a message indicating that access is *not* allowed and signs you out from the computer system. Similarly, if your primary menu has been marked as "out-of-order" (an option attribute), Kernel also denies you access (see Figure 2).

REF: For more information on primary menus, see the "Menu Manager" section.

Figure 2: Access Denied Due to No Primary Menu or Menu "Out of Order" Message

```
ACCESS CODES: ********
VERIFY CODES: ********
No access allowed for this user.
```

2.1.1 Defining a Strong Verify Code

While Access codes are a unique identifier (i.e., username) for your user record in Kernel's NEW PERSON (#200) file, Verify codes are secret passwords assuring that the person signing on is the one for whom the user record was established. You rarely need to be issued a new Access code, but you *must* change your Verify code (i.e., password) if you suspect that someone else has used it to gain access to the system or when your Verify code has expired (i.e., every **90** days or less). You can change your Verify code with the **Edit User Characteristics** [XUEDITSELF] option, which is available from the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu) **User's Toolbox** [XUSERTOOLS] menu.

NOTE: Kernel records all signons to VistA using appropriate user credentials via either of the following methods:

- Access and Verify codes.
- 2-Factor Authentication (2FA)—Digital certificate in a VA-approved smart card, such as the Personal Identification Verification (PIV) smart card plus a Personal Identification Number (PIN).

  Once a user starts using PIV for all access to VistA, their Verify code will expire after **90** days. An expired Verify code will *not* prevent access to VistA through PIV+PIN. If for some reason the user later needs to access VistA with their Access and Verify codes, the first time they sign on with their expired Verify code they will be prompted to reset their Verify code before continuing.

REF: For more information on using the **Edit User Characteristics** [XUEDITSELF] option to reset the Verify code, see the "Edit User Characteristics Option" section.

REF: For more information on Verify code expiration dates, see Section 3.1.2.9, LIFETIME OF VERIFY CODE."

As of Kernel patch XU*8.0*180, *strong* Access and Verify codes *must* adhere to the following criteria:

- Access and Verify codes *cannot* be identical.
- Verify codes (i.e., passwords) *must* be at least **8** characters in length. A *minimum* of **15** characters is *recommended* and may be enforced at a later date.
- Strong passwords in general contain at least three of the following four-character types:
  - Uppercase letters
  - Lowercase letters
  - Numbers

    - Special characters/symbols that are neither letters nor numbers (e.g., -, _, #, &, $, *, @)

      NOTE: The caret (^) is a reserved symbol and *cannot* be used as part of a Verify code. Also, some *non*-VistA-based systems restrict certain special characters/symbols used as part of a username or password.

  Because VistA is case-insensitive, VistA only has three sets of characters from which to build a strong Verify code (i.e., password):

    - Letters (of any case)
    - Numbers
    - Special characters/symbols that are neither letters nor numbers (e.g., -, _, #, &, $, *, @)

      NOTE: Some *non*-VistA-based systems restrict certain special characters/symbols used as part of a username or password.

- Verify codes *must* be changed at least every **90** days (or less). You *must* change your Verify code at periodic intervals as specified by the system administrators. Information systems shall *not* permit re-assignment of the last three passwords used. When required, you are prompted during signon to pick a new Verify code.

  REF: For more information on Verify code expiration dates, see Section 3.1.2.9, LIFETIME OF VERIFY CODE."

- Accounts that have been inactive for **90** days shall be disabled.
- To preclude password guessing, an intruder lockout feature shall suspend accounts after **five** invalid attempts to log on:
    - Where around-the-clock system administration service is available, system administrator intervention shall be required to clear a locked account.
    - Where around-the-clock system administration service is *not* available, accounts shall remain locked out for at least **ten** minutes.

NOTE: These rules are taken from the *VA Account and Password Management Interim Policy* document.

All of these restrictions are enforced whenever Access or Verify codes are created or changed.

These changes were made to meet VA Directive 6500 and VA Handbook 6500.

REF: For more tips and general advice regarding Access and Verify codes and security in general, see the *Kernel Security Tools Manual*.

2.1.1.1 Why Longer Passwords?

Passwords used to access VA systems *must* be at least **8** characters long because longer passwords are stronger, and thus, harder to guess than shorter ones. While VistA currently supports **8**-character passwords (Verify codes), current security policy *recommends* that a minimum of **15** characters be used. This policy will be enforced in a future VistA Kernel patch.

The more tries it takes a hacker or a program to guess a password, the more secure the system is. Adding just one character to the length of a password greatly increases the difficulty of guessing the password.

For an **8**-character password made up of letters and numbers (assuming you can repeat characters and that there are no restrictions, such as requiring the first character to be a letter), there are **36** possibilities for the first position, **36** possibilities for the second position, **36** possibilities for the third position, and so on. Thus, there are **36 x 36 x 36 x 36 x 36 x 36 x 36 x 36 = 2,821,109,907,456** possibilities for an **8**-character password.

If you have forgotten your Verify code (password), the site's Information Security Officer (ISO) should delete the existing Verify code, and then instruct you to sign on again. At the "Verify code" prompt simply press the **<Enter>** key without making any other entries. You are prompted to enter a new Verify code and then re-prompted to enter the same Verify code again as confirmation. If you do *not* want to bother inventing a Verify code, entering a question mark (**?**) at the Verify code prompt displays a possible although cryptic choice (e.g., DKMl&493). Entering a question mark a second time displays another choice. When you log off, you're reminded to remember the new Verify code for use at your next signon.

2.1.2 LOGIN Menu Template

You can execute a script of options on your first signon of the day by having a MENU template called **LOGIN**.

REF: For more information, see the "Menu Manager: User Interface" section.

2.1.3 Signon Shortcuts

In roll-and-scroll VistA, to reach the primary menu in one step at the “ACCESS CODES:” prompt, you can enter the Access and Verify code as one string separated by a semicolon:

Figure 3: Entering the Access and Verify Codes at the Same Time

```
ACCESS CODES: ACCESSCODE;VERIFYCODE
Good afternoon.     You last signed on today at 12:00
```

To “jump start” directly to a particular option, you can specify the name of an option after another semicolon:

Figure 4: Entering the Access and Verify Codes at the Same Time and Jumping Directly to a Specified Option

```
ACCESS CODES: ACCESSCODE;VERIFYCODE;INTRO
Good afternoon.     You last signed on today at 12:00
INTROductory text edit
```

To force the Kernel query of the terminal type identity, you can include a colon anywhere in the string.

REF: If you want to avoid the terminal type query, see the “Terminal Type Prompt” section.

2.1.4 Normal Signoff

When you complete a session on the computer system, you should sign off the system so that no one can come along and use the computer system under your identity. There are several ways you can sign off of the system.

Figure 5: System Commands: Menu Options for Signoff

```
SYSTEM COMMAND OPTIONS                                      [XUCOMMAND]
  Halt                                                         [XUHALT]
  Continue                                                 [XUCONTINUE]
  Restart Session                                            [XURELOG]
```

One way to sign off is to enter “**halt**” at any menu prompt. When you sign off using “**halt**,” at next signon, after entering Access and Verify codes, your normal primary menu is your first menu.

Or, to sign off, you can enter “continue.” At your next signon, after entering Access and Verify codes, your last-used menu when you signed off is your first menu for that session.

If remotely connected via modem or other network device, you can enter "restart" to sign out of Kernel without dropping the communication line.

Finally, you can sign off without using any of these shortcuts simply by pressing **<Enter>** at each menu prompt to step back up the menu pathway and finally exit.

REF: For more information on menus and menu prompts, see the "Menu Manager: User Interface" section.

2.1.5 Abnormal Signoff and Error Handling

If you encounter an error while using the VistA computer system, Kernel traps it, issues the message "Sorry 'bout that", and attempts to return you to your primary menu. Kernel can recover from most error conditions, and given a suitable environment, permits you to continue. Some error conditions, however, cause an abnormal exit, which immediately logs you off the computer system. When this happens, you can sign on again if you still need to use the computer system.

2.1.6 Terminal Type Prompt

When signing on, you may be prompted to enter a terminal type. You should *not* see this prompt very often, however, since Kernel usually can identify your terminal type without needing to prompt you to enter one. If you are prompted, you should enter the name of the actual terminal type to use (e.g., C-VT220). The entered terminal type tells Kernel how to support screen-oriented and other enhanced displays. If unusual circumstances arise and the wrong terminal type is in effect, you can redefine it by using the **Edit User Characteristics** [XUEDITSELF] option (available through the **User's Toolbox** [XUSERTOOLS] menu).

The **Edit User Characteristics** [XUSEREDITSELF] option lets you edit a setting (ASK DEVICE TYPE AT SIGN-ON) that allows you to decide whether to bypass the usual terminal type query. If you always work at the same terminal and want to save a small amount of time during the signon process, you can set ASK DEVICE TYPE AT SIGN-ON to **DON'T ASK**. Kernel then assumes that your last terminal type should be used as the default.

If you have ASK DEVICE TYPE AT SIGN-ON set to **DON'T ASK**, and then sign on using a terminal whose terminal type is different from the one normally used, you should sign on by including a colon (**:**) after your Access code. This forces Kernel to query the terminal for its identity. Alternatively, once signed on, you could invoke the **Edit User Characteristics** [XUSEREDITSELF] option to change your terminal type to the one currently in use. Or, you could use this option to reset the ASK DEVICE TYPE AT SIGN-ON question to **ASK**, log off and sign back on (whereby Signon/Security obtains the correct terminal type identification).

REF: For more information on the **Edit User Characteristics** [XUEDITSELF] option, see the "Edit User Characteristics Option" section.

2.2 Escaping from a Jumbled Screen

One consequence of your signon terminal type *not* matching the actual one being used is that full-screen display could appear jumbled. To escape from a ScreenMan form (e.g., Edit User Characteristics), all you need to do is enter two carets (^), each followed by the **<Enter>** key. To escape from VA FileMan's Screen Editor, you should press **<PF1>E** to exit.

2.3 Alerts

After signing on, you could be presented with an alert notice just before the menu prompt. If so, you need to pick the **View Alerts** [XQALERT] option for viewing alerts to take care of urgent, pending matters.

REF: For more information about alerts, see the "Alerts" section.

Figure 6: System Commands: View Alerts Option

```
SYSTEM COMMAND OPTIONS ...                                      [XUCOMMAND]
  View Alerts "VA"                                                [XQALERT]
```

2.4 User's Toolbox Menu

The **User's Toolbox** [XUSERTOOLS] menu is available from any menu prompt, by entering the toolbox synonym (e.g., "**TBOX**") or "**User's Toolbox**." It makes available, from one menu, some of the most frequently used Kernel options, as shown in Figure 7.

Figure 7: User's Toolbox Menu Options

```
Select User's Toolbox Option:

          Change my Division                                [XUSER DIV CHG]
          Display User Characteristics                         [XUUSERDISP]
          Edit User Characteristics                         [XUSEREDITSELF]
          Electronic Signature code Edit                          [XUSESIG]
          Menu Templates ...                                      [XQTUSER]
          Spooler Menu ...                                    [XU-SPL-MENU]
             **> Locked with XUMGR
          Switch UCI                                        [XU SWITCH UCI]
          TaskMan User                                         [XUTM USER]
          User Help                                           [XUUSERHELP]
```

Table 2 lists the options contained in the **User's Toolbox** [XUSERTOOLS] menu and the sections where each option is described:

Table 2: User's Toolbox Menu Options and Documentation References

Option Text	Section Described
Change my Division [XUSER DIV CHG]	Signon/Security: User Interface
Display User Characteristics [XUUSERDISP]	Signon/Security: User Interface
Edit User Characteristics [XUSEREDITSELF]	Signon/Security: User Interface
Electronic Signature code Edit [XUSESIG]	Electronic Signatures
Menu Templates [XU-SPL-MENU]	Menu Manager: User Interface
Spooler Menu [XU-SPL-MENU] (locked with XUMGR security key)	Spooling
Switch UCI [XU SWITCH UCI]	Signon/Security: User Interface
TaskMan User [XUTM USER]	TaskMan: User Interface
User Help [XUUSERHELP]	(accesses online help)

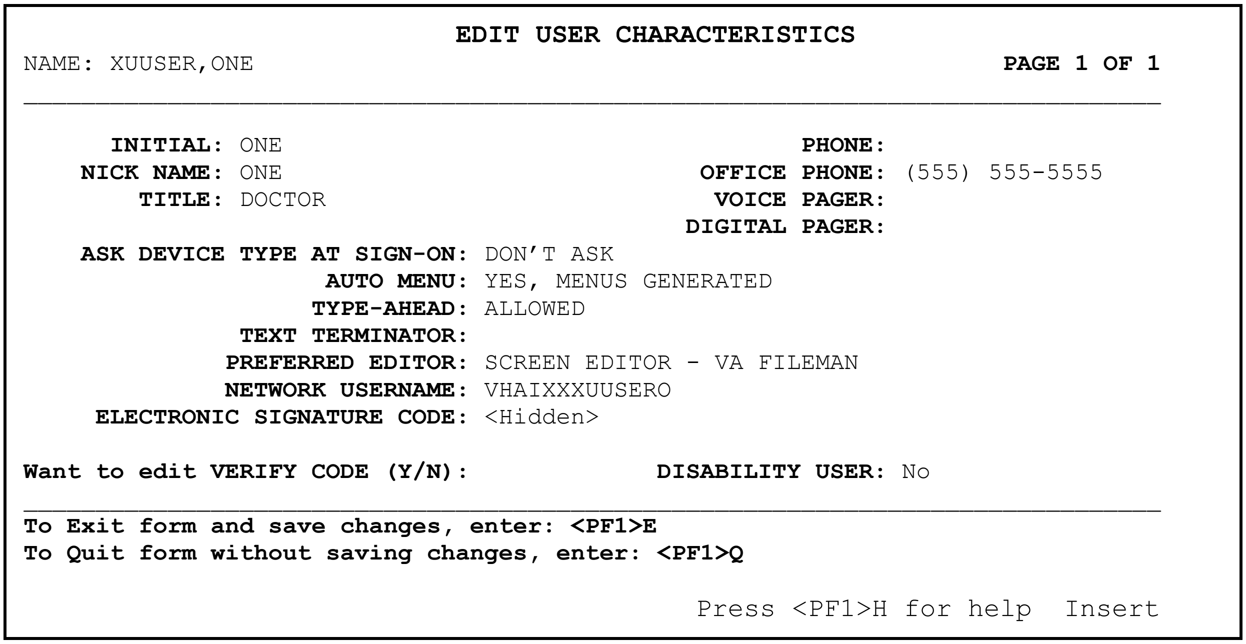
2.5 Change my Division Option

The **Change my Division** [XUSER DIV CHG] option allows users to select from a list of divisions, if any, stored for that user in the NEW PERSON (#200) file.

2.6 Edit User Characteristics Option

The **Edit User Characteristics** [XUSEREDITSELF] option is one of the options available from the **User's Toolbox** [XUSERTOOLS] menu. It allows you to define some characteristics of your online environment via ScreenMan, as shown in Figure 8:

Figure 8: Edit User Characteristics Option—ScreenMan Form



```
                              EDIT USER CHARACTERISTICS
NAME: XUUSER,ONE                                                    PAGE 1 OF 1
_______________________________________________________________________________

      INITIAL: ONE                                    PHONE:
    NICK NAME: ONE                             OFFICE PHONE: (555) 555-5555
        TITLE: DOCTOR                           VOICE PAGER:
                                              DIGITAL PAGER:
    ASK DEVICE TYPE AT SIGN-ON: DON'T ASK
                     AUTO MENU: YES, MENUS GENERATED
                    TYPE-AHEAD: ALLOWED
               TEXT TERMINATOR:
              PREFERRED EDITOR: SCREEN EDITOR - VA FILEMAN
              NETWORK USERNAME: VHAIXXXUUSERO
     ELECTRONIC SIGNATURE CODE: <Hidden>

Want to edit VERIFY CODE (Y/N):             DISABILITY USER: No
_______________________________________________________________________________
To Exit form and save changes, enter: <PF1>E
To Quit form without saving changes, enter: <PF1>Q

                                               Press <PF1>H for help    Insert
```

Table 3 lists a number of NEW PERSON (#200) file field values that you can edit with the **Edit User Characteristics** [XUEDITSELF] option:

Table 3: Edit User Characteristics Option—Editable Fields

Field	Description
INITIAL (#1)	Enter your initials, which can serve as an alternate way for users to specify your account (e.g., when sending mail to you).
NICK NAME (#13)	Enter a nick name, which can serve as an alternate way for users to specify your account (e.g., when sending mail to you).
TITLE (#8)	Enter a title from a given list of choices or enter a new TITLE.
Telephone Contact Information: • PHONE (HOME) (#.131) • OFFICE PHONE (#.132)	Enter the appropriate phone numbers in the fields indicated.

Field	Description
- VOICE PAGER (#.137) - DIGITAL PAGER (#.138)	
ASK DEVICE TYPE AT SIGN-ON (#200.05)	This field controls whether Kernel should determine what kind of terminal you are using when you sign on. If this is set to DON'T ASK, Kernel assumes you are using the same kind of terminal you used the last time you signed on. This can cause problems if you are using a different kind of terminal (screen displays may *not* work properly), so this should normally be set to ASK.
AUTO MENU (#200.06)	This field determines whether, in the menu system, a list of items on the current menu is displayed with the menu prompt. Beginning users should usually set AUTO MENU to **YES**, so that they can see menu items for each menu. Experienced users who are familiar with their menus may prefer to set this field to **NO**, which makes menu displays speedier, since individual items on each menu are *not* displayed.
TYPE-AHEAD (#200.09)	This field controls whether characters you type faster than the system can process end up being processed or not. Normally you should set TYPE-AHEAD to **YES**, so that keystrokes you enter are *not* lost due to system slowness.
TEXT TERMINATOR (#31.2)	The TEXT TERMINATOR is a setting used by VA FileMan's Line Editor. When you are using the Line Editor and are importing text from an external source, you may *not* want a blank line to indicate the end-of-file, which could prematurely terminate the text transfer. By default, the TEXT TERMINATOR in VA FileMan's Line Editor is the carriage return character (**<Enter>**). Setting this to another character string, like **ZZ** (something that is *not* encountered in the target text) can permit downloading without interruption. If you change the setting of the TEXT TERMINATOR from the default of the carriage return character, you need to remember your TEXT TERMINATOR when using the Line Editor; otherwise, you are unable to exit the Line Editor. **REF:** For more information on the TEXT TERMINATOR, see the *VA FileMan User Manual*.
PREFERRED EDITOR (#31.3)	Users can choose which text editor Kernel uses when you edit word-processing fields on the system. You can choose any editor defined on your system.

Field	Description
NETWORK USERNAME (#501.1)	Enter your network user name. This is the username that is used by the Windows Active Directory (AD). It allows VISN data extracts to link the VistA user with their network user name. Format: “VHA” + 3-character station ID + first 5 characters of last name + first character of first name For example, for user One Xuuser at Station ID 999, the network user name would be: VHA999XUUSEO Holders of the XUMGR security key can override this field. **NOTE:** This field was added to the NEW PERSON (#200) file with Kernel patch XU*8.0*514.
ELECTRONIC SIGNATURE CODE (#20.4)	Enter a new electronic signature code. This is a code (similar to a password) used to electronically sign documents within VistA. When you press **Enter**, the code is hidden for security purposes.
VERIFY CODE (#7.2)	Users can change their VERIFY CODE by answering **YES** to this field. First enter your current VERIFY CODE; then, enter a new VERIFY CODE. You are asked to confirm the new VERIFY CODE by entering it a second time; if you confirm it, the new VERIFY CODE takes effect immediately.
DISABILITY USER (#508.1)	Users can set this field value to **Yes** or **No**. If the user has a visual disability, set this field to **Yes**. Applications should check this field value for visually disabled users for Section 508 compliance. **NOTE:** This field was added to the NEW PERSON (#200) file with Kernel Patch XU*8.0*741. The **DISABILITY USER** display field was added to the “**EDIT USER CHARACTERISTICS**” screen with Kernel Patch XU*8.0*753.

2.7 Display User Characteristics Option

The **Display User Characteristics** [XUUSERDISP] option, like the **Edit User Characteristics** [XUSEREDITSELF] option, is an option in the **User's Toolbox** [XUSERTOOLS] menu. It prints out a description of many of the characteristics of your current computing environment, including some of the characteristics that can be set through the **Edit User Characteristics** [XUSEREDITSELF] option.

Figure 9: Display User Characteristics Option—Sample Output and User Dialog

```
XUUSER,TWO  (#9999)  DEVICE: DEVICE: TELNET   ($I: TNA730:)      JOB: 541754169

ENVIRONMENT                             ATTRIBUTES
-----------                             -----------
   Site ........ TESTSITE                Type-ahead ....... Y
   UCI ......... KRN,KDE                Time-out ......... 300
   Signed on ... 08:48                  Fileman code(s) .. #
   Terminal type C-VT100

Person Class: Physicians (M.D. and D.O.)
                Physician/Osteopath
                  Pathology, Anatomic

KEYS HELD
---------
XMMGR                XUPROG              XUPROGMODE

MENU PATH
---------
  SYSTEM COMMAND OPTIONS (XUCOMMAND)
    User's Toolbox (XUSERTOOLS)
      Display User Characteristics (XUUSERDISP)

'^' to escape, <CR> to view Mailman user info: <Enter>

Current Banner: Technical Writer
Last used MailMan: 07/12/06@15:09
NEW messages: 274 (274 in the IN basket)

Office phone:  (555) 555-5555
Fax:           (555) 555-5555
Add'l phone:   (555) 555-5555
Add'l phone:   (555) 555-5555

Introduction:
  My name is One Xmuser and I am one of the Technical Writers for the
  Common Services (CS) products/projects (e.g., Broker, Components,
  Kernel, VA FileMan, MailMan, Toolkit).

Mail Groups:
  REDACTED STAFF                               (Public)
  KERNEL PROGRAMMERS                           (Public)
```

2.8 Switch UCI Option

The **Switch UCI** [XU SWITCH UCI] option allows users to select from a list of UCIs, if any, stored for that user in the NEW PERSON (#200) file.

2.9 Summary

VistA's Kernel's Signon/System Security module provides the means for signing into Kernel with a unique identity. Once you complete the signon process, you are sent to Kernel's menu system, where you can run any option your system manager has placed in your menus. When you finish a computer session, always be sure to sign off; this protects your account from misuse by someone else.

3 Signon/Security: System Management

This section describes the system management tools for Kernel's Signon/Security module.

3.1 Signon Process

If signons are enabled, as shown in the Signon Flow Chart in Figure 13, the signon process begins with a gathering of information from the KERNEL SYSTEM PARAMETERS (#8989.3) file and then from the DEVICE (#3.5) file to determine whether to allow signon for this session and, if so, how to create an appropriate environment. If, for example, the MAX SIGNON ALLOWED limit has been reached, the signon attempt fails. If the current device is tied to a routine (as specified in the TIED ROUTINE field of the DEVICE [#3.5 file]), that routine is executed, and the session is halted. If *not*, the user is prompted for Access and Verify codes. After a successful signon, attributes for that user are then retrieved from the NEW PERSON (#200) file. Signon/Security then sends the user to Menu Manager. If a primary menu is associated with the device (PRIMARY MENU OPTION field in the DEVICE [#3.5] file), that menu is presented. Otherwise, the user's primary menu is presented. If the user does *not* have a primary menu (the PRIMARY MENU OPTION field in the NEW PERSON [#200] file is **NULL**), the session is halted.

The signon flow chart in this section (see Figure 13) illustrates the procedural steps taken by Kernel's Signon/Security system to determine whether to permit signons and, if so, how to create an appropriate computing environment. Typically, after site parameters and device characteristics are checked:

1. System prompts the user for their Access and Verify codes. Alternatively, client applications that are enabled to use 2-Factor Authentication (2FA) will automatically enter a Security Assertion Mark-up Language (SAML) token obtained from Identity and Access Management (IAM) instead of an Access and Verify code to authenticate and identify the user.
2. System collects user attributes.
3. System presents a primary menu prompt to the user.

3.1.1 Introductory Text

Before gathering system parameters or prompting for Access and Verify codes, Signon/Security displays contents of the INTRO TEXT field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. The text can be edited with the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option or with the **Introductory text edit** [XUSERINT] option, an option specially designed for this purpose).

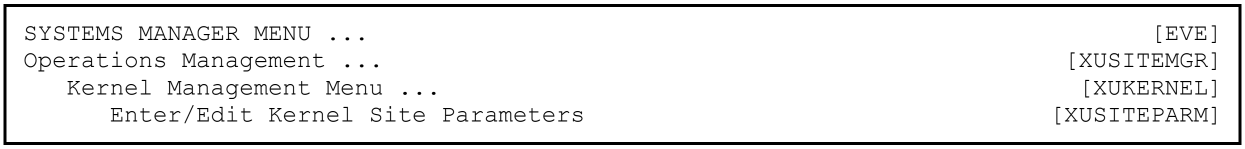
Figure 10: Introductory text edit Option

```
SYSTEMS MANAGER MENU ...                                        [EVE]
Operations Management ...                                 [XUSITEMGR]
   Introductory text edit                                  [XUSERINT]
```

3.1.2 Parameters Checked during Signon

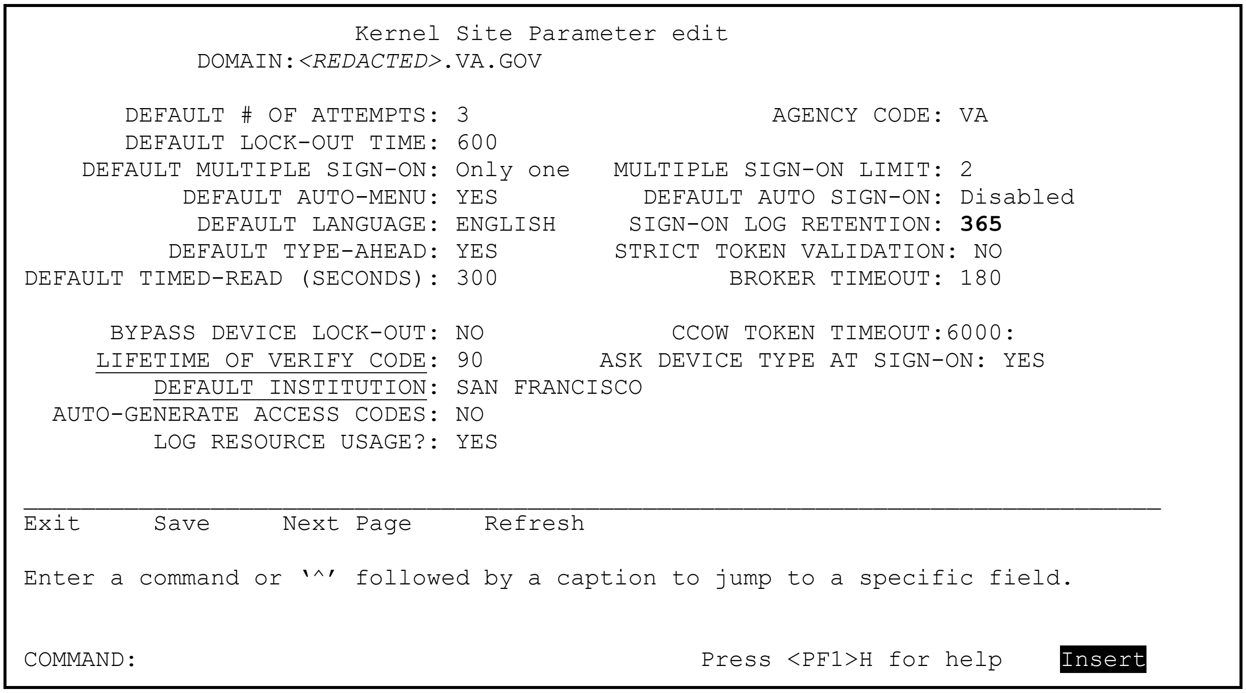
Various parameters are checked as an initial step in the signon process. The KERNEL SYSTEM PARAMETERS (#8989.3) file stores the default values for most of the parameters. Values for critical fields should be defined by system administrators when Kernel is installed. The values in the KERNEL SYSTEM PARAMETERS (#8989.3) file can be edited any time, though, with the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option.

Figure 11: Enter/Edit Kernel Site Parameters Option



```
SYSTEMS MANAGER MENU ...                                                 [EVE]
Operations Management ...                                          [XUSITEMGR]
   Kernel Management Menu ...                                       [XUKERNEL]
      Enter/Edit Kernel Site Parameters                           [XUSITEPARM]
```

Figure 12: Enter/Edit Kernel Site Parameters Option—ScreenMan Form 1



```
                       Kernel Site Parameter edit
            DOMAIN:<REDACTED>.VA.GOV

       DEFAULT # OF ATTEMPTS: 3                   AGENCY CODE: VA
       DEFAULT LOCK-OUT TIME: 600
    DEFAULT MULTIPLE SIGN-ON: Only one   MULTIPLE SIGN-ON LIMIT: 2
           DEFAULT AUTO-MENU: YES          DEFAULT AUTO SIGN-ON: Disabled
            DEFAULT LANGUAGE: ENGLISH     SIGN-ON LOG RETENTION: 365
          DEFAULT TYPE-AHEAD: YES        STRICT TOKEN VALIDATION: NO
DEFAULT TIMED-READ (SECONDS): 300                BROKER TIMEOUT: 180

      BYPASS DEVICE LOCK-OUT: NO             CCOW TOKEN TIMEOUT:6000:
     LIFETIME OF VERIFY CODE: 90        ASK DEVICE TYPE AT SIGN-ON: YES
         DEFAULT INSTITUTION: SAN FRANCISCO
  AUTO-GENERATE ACCESS CODES: NO
         LOG RESOURCE USAGE?: YES

_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

3.1.2.1 Signon Attempts and Device Lock-out Times

The DEFAULT # OF ATTEMPTS field in the KERNEL SYSTEM PARAMETERS (#8989.3) file holds the default limit of the number of times a user can try to enter a valid Access and Verify code pair. When the limit is reached, Signon/Security is unresponsive for the duration specified by the DEFAULT LOCK-OUT TIME field. The values for number of attempts and lock-out time are overridden by any values for the current device specified by comparable fields in the DEVICE (#3.5) file.

Device values are ignored, however, if the BYPASS DEVICE LOCK-OUT site parameter in the KERNEL SYSTEM PARAMETERS (#8989.3) file is set to **YES**. In particular, the fields that are bypassed are:

- OUT-OF-SERVICE DATE
- SECURITY
- PROHIBITED TIMES FOR SIGN-ON

Device values are put back into effect for the current device if the DEVICE file's PERFORM DEVICE CHECKING field is set to **YES**.

3.1.2.2 MAX SIGNON ALLOWED

One Kernel site parameter used in the initial signon screening is MAX SIGNON ALLOWED. It is a field within the VOLUME SET Multiple field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. Its value sets an upper limit for number of M processes (interactive, background, and system) that can run concurrently on the specified Volume Set or CPU. The TASKMAN JOB LIMIT, a field in the TASKMAN SITE PARAMETERS (#14.7) file, should be set to a number slightly lower than MAX SIGNON ALLOWED to leave room for a few interactive logons when TaskMan is busiest.

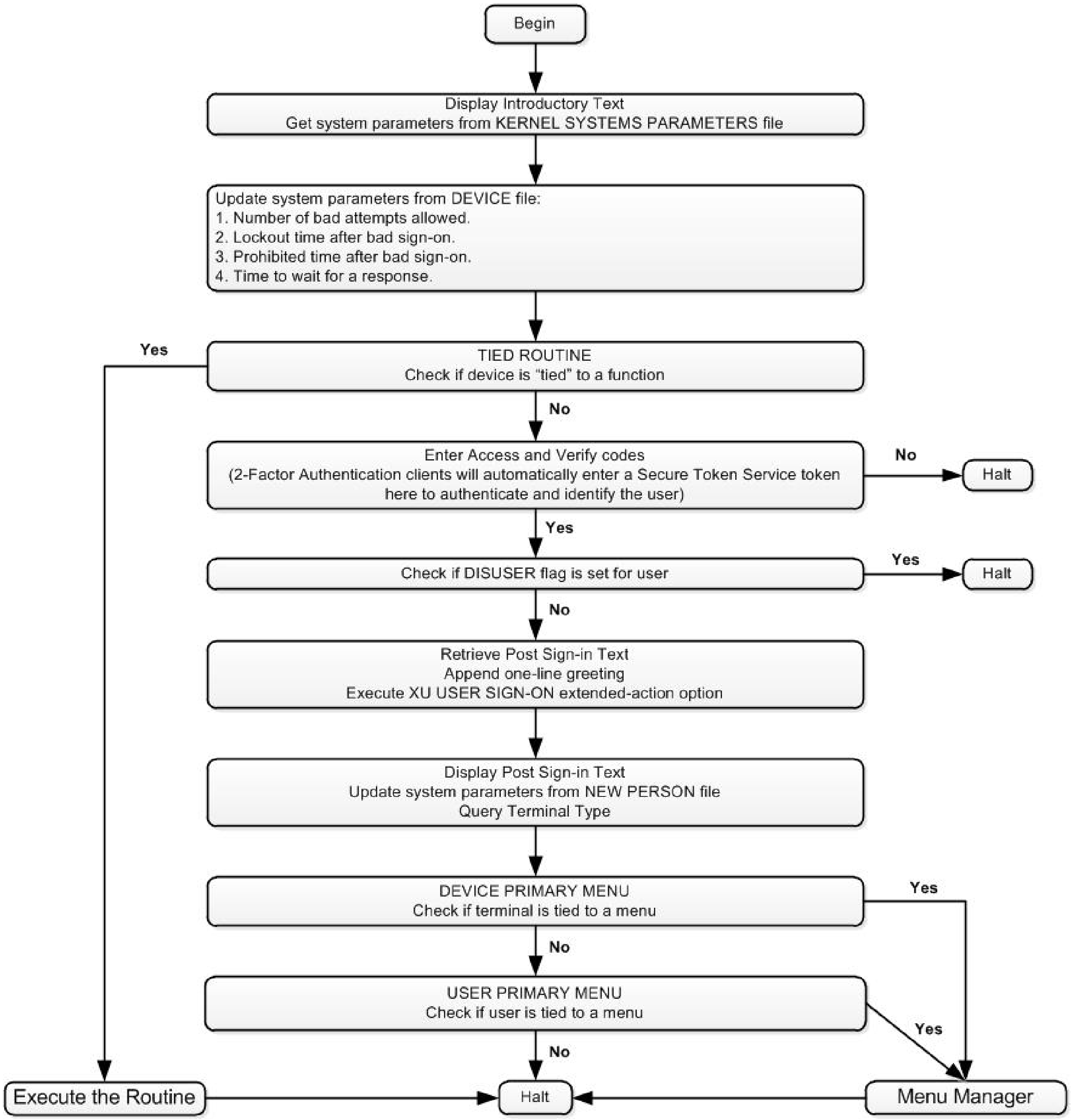
NOTE: OpenVMS Sites: The OpenVMS interactive logins parameter (set by the DCL command **SET LOGINS/INTERACTIVE**) should be set to a number less than the Kernel MAX SIGNON ALLOWED to conserve system resources. If the OpenVMS limit is set too high in relation to the Kernel limit, users try to access Kernel only to be rejected when reaching Signon/Security. That means that they would waste system resources by creating a new OpenVMS process and activating a Caché image, all to no avail.

REF: For more information about alerts, see "Alerts."

3.1.2.3 PROHIBITED TIMES FOR SIGN-ON

Time periods can be specified, during which interval signons can be barred by device or by user. This is controlled by the PROHIBITED TIMES FOR SIGN-ON field in the DEVICE (#3.5) file and a comparable field in the NEW PERSON (#200) file.

Figure 13: Kernel Signon Flow Chart


Begin
Display Introductory Text
Get system parameters from KERNEL SYSTEMS PARAMETERS file
Update system parameters from DEVICE file:
1. Number of bad attempts allowed.
2. Lockout time after bad sign-on.
3. Prohibited time after bad sign-on.
4. Time to wait for a response.
Yes
TIED ROUTINE
Check if device is "tied" to a function
No
Enter Access and Verify codes
(2-Factor Authentication clients will automatically enter a Secure Token Service token here to authenticate and identify the user)
No
Halt
Yes
Check if DISUSER flag is set for user
Yes
Halt
No
Retrieve Post Sign-in Text
Append one-line greeting
Execute XU USER SIGN-ON extended-action option
Display Post Sign-in Text
Update system parameters from NEW PERSON file
Query Terminal Type
DEVICE PRIMARY MENU
Check if terminal is tied to a menu
Yes
No
USER PRIMARY MENU
Check if user is tied to a menu
Yes
No
Execute the Routine
Halt
Menu Manager

3.1.2.4 Multiple Sign-On Restriction

The DEFAULT MULTIPLE SIGN-ON field in the KERNEL SYSTEM PARAMETERS (#8989.3) file controls whether users can create two or more simultaneous sessions by signing on to more than one device. The setting is overridden by comparable fields in the DEVICE (#3.5) and NEW PERSON (#200) files, respectively. The value is checked at signon to prevent unauthorized multiple sessions.

If multiple signons are prohibited, problems can occur if users experience an abnormal exit such that the signon record *cannot* be cleared. To clear an individual user, use the **Release user** [XUSERREL] option. To make sure all users are clear when the system is brought up after a crash, system administrators can use the **Clear all users at startup** [XUSER-CLEAR-ALL] option.

3.1.2.5 INTERACTIVE USER'S PRIORITY

The INTERACTIVE USER'S PRIORITY parameter in the KERNEL SYSTEM PARAMETERS (#8989.3) file should usually be left **NULL**. A setting here affects the job priority of interactive users and could result in poor response time.

3.1.2.6 ASK DEVICE TYPE AT SIGN-ON

The ASK DEVICE TYPE AT SIGN-ON parameter controls whether the user's current device at signon is queried for its display attributes (**DA**). Thus, the correct terminal type can be identified without prompting the user.

It is *recommended* that ASK DEVICE TYPE AT SIGN-ON be set to **ASK** so that Signon/Security performs the **DA** query and allows the Device Handler to set up the correct terminal type attributes. This has become more important with the advent of screen control. VA FileMan's Screen Editor and Screen Manager, for example, does *not* function properly if the terminal type recorded by Kernel fails to match the actual terminal type being used.

As with other parameters, the site default (ASK DEVICE TYPE AT SIGN-ON field in the KERNEL SYSTEM PARAMETERS [#8989.3] file) is overridden by a **DON'T ASK** setting for the device (like-named field in the DEVICE [#3.5] file), which would similarly be overridden by a **DON'T ASK** setting for the user (like-named field in the NEW PERSON [#200] file). A **NULL** value functions as **ASK**. The user override can be set by any user via the **Edit User Characteristics** [XUSEREDITSELF] option.

REF: For more information on the **Edit User Characteristics** [XUEDITSELF] option, see the "Edit User Characteristics Option" section.

If the parameter is set to **DON'T ASK**, Signon/Security does *not* perform the **DA** query and assumes the user's last terminal type is still appropriate. Although the difference in resource consumption is negligible, the user can appreciate a split second's savings in time. Thus, bypassing the **DA** query can be acceptable, if the same terminal type is always being used. But if the user should sign onto another terminal type, problems can occur with the presentation of screen-oriented displays unless the user knows how to change the terminal type to match the actual current one.

If the device is *non*-ANSI-standard, Signon/Security may *not* find a **DA** but continues to determine the terminal's identity by querying its answerback message. All known *non*-ANSI devices (e.g., Qume 102 terminal) should have their answerback messages programmed. This is accomplished by using the terminal type setup mechanism and entering **C-QUME** as the Qume 102's answerback message. The name *must* match an entry in Kernel's TERMINAL TYPE (#3.2) file to take effect. If the answerback message contains additional characters (e.g., a serial number), the message does *not* match an entry in the TERMINAL TYPE (#3.2) file and is useless for signon purposes.

If the terminal's **DA** return code does *not* match an entry in the DA RETURN CODES (#3.22) file, or if the terminal is *non*-ANSI and *cannot* be programmed with an appropriate answerback message, Signon/Security prompts the user to identify the terminal type if the user's ASK DEVICE TYPE AT SIGN-ON setting is set to **ASK**. This is the only case in which the terminal type prompt is asked during signon. The last terminal type used is presented as the default (it is stored in the NEW PERSON [#200] file). If ASK DEVICE TYPE AT SIGN-ON is set to **DON'T ASK**, Signon/Security assumes that the last terminal type is appropriate and does *not* prompt the user for validation.

3.1.2.7 Display Attributes (DA) Return Codes

The DA RETURN CODES (#3.22) file is used to equate **DA** return codes to entries in the TERMINAL TYPE (#3.2) file. You can use the **DA Return Code Edit** [XU DA EDIT] option to automate the population of the DA RETURN CODES (#3.22) file.

REF: For more information, see the "Managing Display Attributes (DA) Return Codes" section in the "Device Handler: System Management" section.

3.1.2.8 SELECTABLE AT SIGNON

System administrators can also control which devices can be selected at signon with a field in the TERMINAL TYPE (#3.2) file. The SELECTABLE AT SIGN-ON flag should be set to **YES** for all devices commonly used for signon. Ordinarily, it should *not* be set for printers (e.g., **P-** terminal types **P-DEC** or **P-OTHER**). To allow the loading of ScreenMan forms and proper functioning of other screen-oriented displays, the flag should also *not* be set for **PK-** types, that is, printers with keyboards. This is *not* an actual restriction, however, but a recommendation.

3.1.2.9 LIFETIME OF VERIFY CODE

To insure that users change their Verify codes at periodic intervals, system administrators should set the LIFETIME OF VERIFY CODE parameter in the KERNEL SYSTEM PARAMETERS (#8989.3) file to a certain number of days. The maximum number is **90** days and the minimum number is **1** day. Thus, sites can choose any number from **1-90** days before requiring users to change their Verify code. At the end of that period (e.g., every **90** days), users *must* then change their Verify codes. Signon/Security checks whether the Verify code needs to be changed, and if so, prompts the user at signon to enter a new Verify code.

3.1.2.10 AUTO-GENERATE ACCESS CODES

When assigning Access codes, the security officer or system administrators can invent an alphanumeric string or can ask Kernel to generate one. If the AUTO-GENERATE ACCESS CODES site parameter in the KERNEL SYSTEM PARAMETERS (#8989.3) file is set to **YES**, only generated, cryptic codes can be assigned. It is *not* necessary to pick the first one presented; others can be generated for selection.

3.1.2.11 DEFAULT INSTITUTION and AGENCY

The institution running Kernel software is defined during the Kernel installation when prompted for the DEFAULT INSTITUTION in the KERNEL SYSTEM PARAMETERS (#8989.3) file. This field is a pointer to the INSTITUTION (#4) file. One or more institutional affiliations can also be associated with a user (e.g., a VA Outpatient Clinic and an Army Medical Center). This data is stored in the DIVISION Multiple field in the NEW PERSON (#200) file. If a user is associated with more than one institution (division), the user is prompted at signon to select a division. In this way, the local variable **DUZ(2)** can be set to the appropriate value. If the user's DIVISION Multiple field is **blank**, the DEFAULT INSTITUTION field (File #8989.3) is used to define **DUZ(2)**. Since the INSTITUTION (#4) file contains a pointer to the AGENCY (#4.11) file, the signed-on user's agency affiliation can also be determined.

The KERNEL SYSTEM PARAMETERS (#8989.3) file also contains the AGENCY CODE (#9). This field is *not* a pointer but is instead a SET OF CODES (e.g., **N** for Navy or **V** for VA). This field is presented for editing during Kernel installation. Its value is used at sign on to set the **DUZ("AG")** local variable. Thus, the agency associated with the overall Kernel system can be determined.

3.1.2.12 AUTO MENU

The AUTO MENU flag, stored in the local variable **DUZ("AUTO")**, is used by Menu Manager to control whether all items on a menu are presented automatically after each cycle through the menu system. If the items are *not* displayed, the user can always invoke the display by entering a question mark (**?**). New users often like to see all the menu choices. Experienced users probably do *not* need to see the choices and the display can be suppressed to save system resources. The user setting for AUTO MENU in the NEW PERSON (#200) file overrides any comparable device setting in the DEVICE (#3.5) file, which will, in turn, override the site parameter default in the KERNEL SYSTEM PARAMETERS (#8989.3) file. Users can edit the setting with the **Edit User Characteristics** [XUSEREDITSELF] option.

REF: For more information on the **Edit User Characteristics** [XUEDITSELF] option, see the "Edit User Characteristics Option" section.

3.1.2.13 TYPE-AHEAD

If TYPE-AHEAD is disabled, any keystrokes that the user enters while computer system processes previously issued instructions do *not* register. If TYPE-AHEAD is enabled, keystrokes entered in advance of processing are stored in the TYPE-AHEAD buffer and is interpreted when the earlier process is finished. New users may experience unwanted results if TYPE-AHEAD is enabled and they had *not* anticipated the effect. Experienced users may prefer TYPE-AHEAD for efficiency. The user setting overrides the device setting, which, in turn, overrides the site parameter setting. Users can edit the setting with the **Edit User Characteristics** [XUSEREDITSELF] option.

> **REF:** For more information on the **Edit User Characteristics** [XUEDITSELF] option, see the "Edit User Characteristics Option" section.

3.1.2.14 TIMED READ

The value for the TIMED READ parameter is stored in the local variable **DTIME** and is used to calculate how long Kernel should wait before terminating a **READ**. If, for example, a user does *not* respond to a menu prompt in the number of seconds defined by the TIMED READ, Kernel takes steps towards signoff and, without subsequent user response, halts the user session. The user setting overrides the device setting, which, as usual, overrides the site default.

3.1.2.15 POST SIGN-IN MESSAGE

The POST SIGN-IN MESSAGE is similar to introductory text (i.e., INTRO TEXT field in File #8989.3), except that Kernel displays it only after a successful signon. Like the introductory text, you can edit the message text using the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option; alternately, you can use the **Post sign-in Text Edit** [XUSERPOST] option, which is specially designed for this purpose:

Figure 14: Post Sign-in Text Edit Option

```
SYSTEMS MANAGER MENU ...                                               [EVE]
Operations Management ...                                        [XUSITEMGR]
   Post sign-in Text Edit                                        [XUSERPOST]
```

Applications can append information to the POST SIGN-IN MESSAGE (on a per-user, per signon basis only) by attaching to the **User sign-on event** [XU USER SIGN-ON] option.

> **REF:** For more information on the **User sign-on event** [XU USER SIGN-ON] option, see the "Signon/Security: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

3.1.2.16 2-Factor Authentication (2FA)

The KERNEL SYSTEM PARAMETERS (#8989.3) file also contains fields that are required to enable 2-Factor Authentication (2FA). These fields are *not* included in the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option, because they should *not* be edited in VA Production systems. If VistA is being installed in a *non*-VA environment, they can be edited using VA FileMan.

Kernel Patch XU*8.0*701 introduced the STRICT TOKEN VALIDATION (#220) field that defaults to **NO**. This field is included in the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option.

NOTE: The original implementation of 2FA did *not* enforce all the verifications of the SSOi token. Thus, the default value of **NO** in this field permits the continued use of "*non*-strict" validation and records any verification failures as "warnings" in the SIGN-ON LOG (#3.081) file. In the future, it will be possible to enforce strict token validation by changing this field's value to **YES**.

NOTE: Kernel Patch XU*8.0*701 also introduced the **cache.cer** certificate file, which contains the chain of Certificate Authorities (CA) used to validate the signature of the 2FA SSOi token. The installation of the **cache.cer** file is *not* mandatory when setting the STRICT TOKEN VALIDATION (#220) field to **NO**; however, the certificate file is required when setting it to **YES**.

2FA Field descriptions (listed in field number order):

- **SECURITY TOKEN SERVICE (#200.1):** When using brokered authentication with a security token issued by a Security Token Service (STS), this field contains the identification of the issuer of the token. The STS is trusted by both the client and the server to provide the interoperable security tokens. Security Assertion Markup Language (SAML) tokens are standards-based XML tokens that are used to exchange security information, including:
    - Attribute statements
    - Authentication decision statements
    - Authorization decision statements

    They can be used as part of a Single Sign-On (SSO) solution allowing a client to talk to services running on disparate technologies. The value of this field should be set to the domain name of the STS as found in the "Issued to:" field of the STS PKI certificate used to digitally sign the token. For VA Production systems, the value should be set to the following value:

    ***<REDACTED>*.va.gov**

- **ORGANIZATION (#200.2):** Identity and Access Management field used to identify the VistA instance organization. For internally authenticated users, this field matches the SUBJECT ORGANIZATION (#205.2) field of the user identified in the NEW PERSON (#200) file. For VA Production systems, this field should always contain the following value:

  Department of Veterans Affairs

- **ORGANIZATION ID (#200.3):** Identity and Access Management field used to uniquely identify the VistA instance organization. For internally authenticated users, this field matches the SUBJECT ORGANIZATION ID (#205.3) field of the user identified in the NEW PERSON (#200) file. For VA Production systems, this field should always contain the following value:

  urn:oid:2.16.840.1.113883.4.349

- **STRICT TOKEN VALIDATION (#220):** This field is used to apply strict credential token validation by Kernel during sign-on. The default is **NO**, *non*-strict token validation. Setting it to **YES**, strict token validation, may cause problems with users signing on to VistA if the required infrastructure is *not* properly set up.

3.1.3 XU USER SIGN-ON Option

The **User sign-on event** [XU USER SIGN-ON] option can attach action-type options to this extended-action-type option, so that software-specific actions can be performed at signon.

REF: For more information, see the “Signon/Security: Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*.

3.1.4 XU USER START-UP Option

The **User start-up event** [XU USER START-UP] option is a protocol option used exclusively during a VistA user signon event. Items attached to this option are “TYPE: action” options in the OPTION (#19) file, which can be used for software-specific actions that prompt users for input upon VistA signon before their primary menu option is displayed. Unlike the **User sign-on event** [XU USER SIGN-ON] option, it can provide interactive prompting to users. It is *not* used for GUI signon. It is called from the **XQ12** routine.

REF: This option was added with Kernel patch XU*8.0*593. For more information, see the “Signon/Security: Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*.

3.1.5 Clear all users at startup Option

Figure 15: Clear All Users at Startup Option

```
PARENT OF QUEUABLE OPTIONS ...                                  [ZTMQUEUABLE OPTIONS]
  Clear all users at startup                                         [XUSER-CLEAR-ALL]
```

If multiple signons are prohibited, users may be prevented from signing on to the system when it is brought up after a crash (which can cause numerous abnormal exits). To prevent this problem from occurring, system administrators can use the **Clear all users at startup** [XUSER-CLEAR-ALL] option. Kernel *recommends* this option be scheduled to run at system startup. Although this option can be invoked interactively without ill effects, it was designed as a background process, thus, it is placed along with other tasked options on the **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu.

REF: For information on how to release a single user, see the "Proxy (Connector) Detail Report Option" section.

3.1.6 Enabling and Disabling Logons

System administrators have full control over whether logons are enabled. Access to a particular Volume Set can be disabled by setting the INHIBIT LOGONS? flag in the VOLUME SET (#14.5) file. Setting the flag to **YES** sets the **^%ZIS("14.5","LOGON","*volume set*")** node, whose presence disallows user logons. That is, logons through Signon/Security, invoking the **^ZU** routine, fails (terminals for user access are usually linked to **ZU** within the operating system setup. Some special terminals, like the console, are untied.) The **^%ZIS("14.5","LOGON","*volume set*")** node is also checked after each cycle through the menu system; signed-on users are logged off as soon as they return to a menu prompt.

3.2 Adding New Users

Creating a new user account involves:

- Adding a record to the NEW PERSON (#200) file.
- Assigning an Access code.
- Assigning a primary menu.

You need the XUMGR security key to assign primary menu options. Even the at-sign (**@**; Programmer access) is insufficient, as checked by the PRIMARY MENU OPTION (#201) field's input transform.

Figure 16: User Management Menu Options: Associated Menu Options when Adding a New User

```
SYSTEMS MANAGER MENU ...                                                [EVE]
User Management ...                                                   [XUSER]
   Add a New User to the System                                    [XUSERNEW]
   Grant Access by Profile <locked: XUMGR>                         [XUSERBLK]
   User Inquiry                                                    [XUSERINQ]
```

3.2.1 Add a New User to the System Option

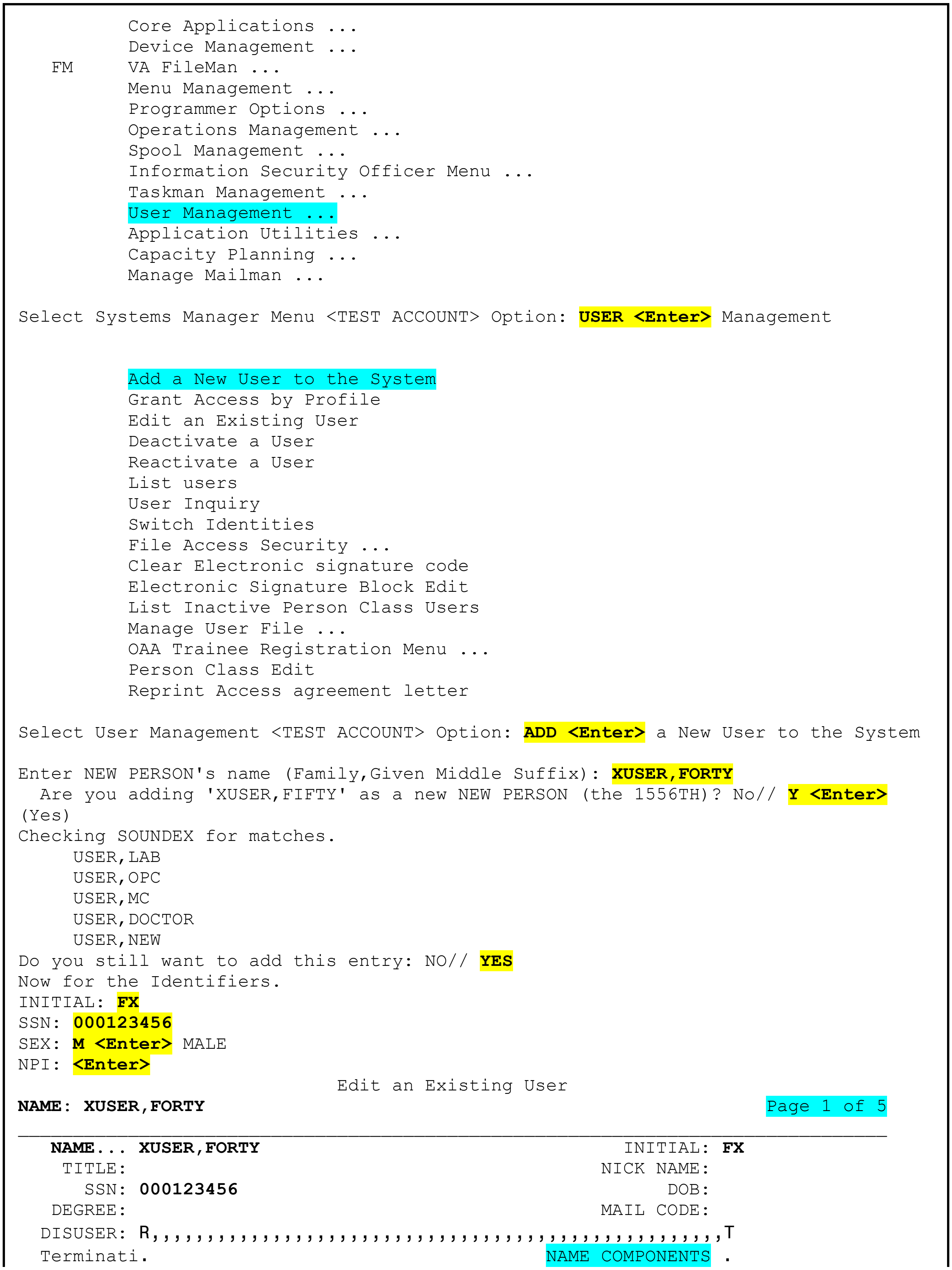
You can use the **Add a New User to the System** [XUSERNEW] option to set up user accounts one-by-one. The **Add a New User to the System** [XUSERNEW] option presents a standard scrolling-mode editing sequence for user attributes.

When using this option, entry of a social security number in the SSN (#9) field is usually required. While SSN is *not* required in the NEW PERSON (#200) file data dictionary, it is a required field when using this option. If the option is used by someone who holds the XUSPF200 security key, however, entry of an SSN is *not* required.

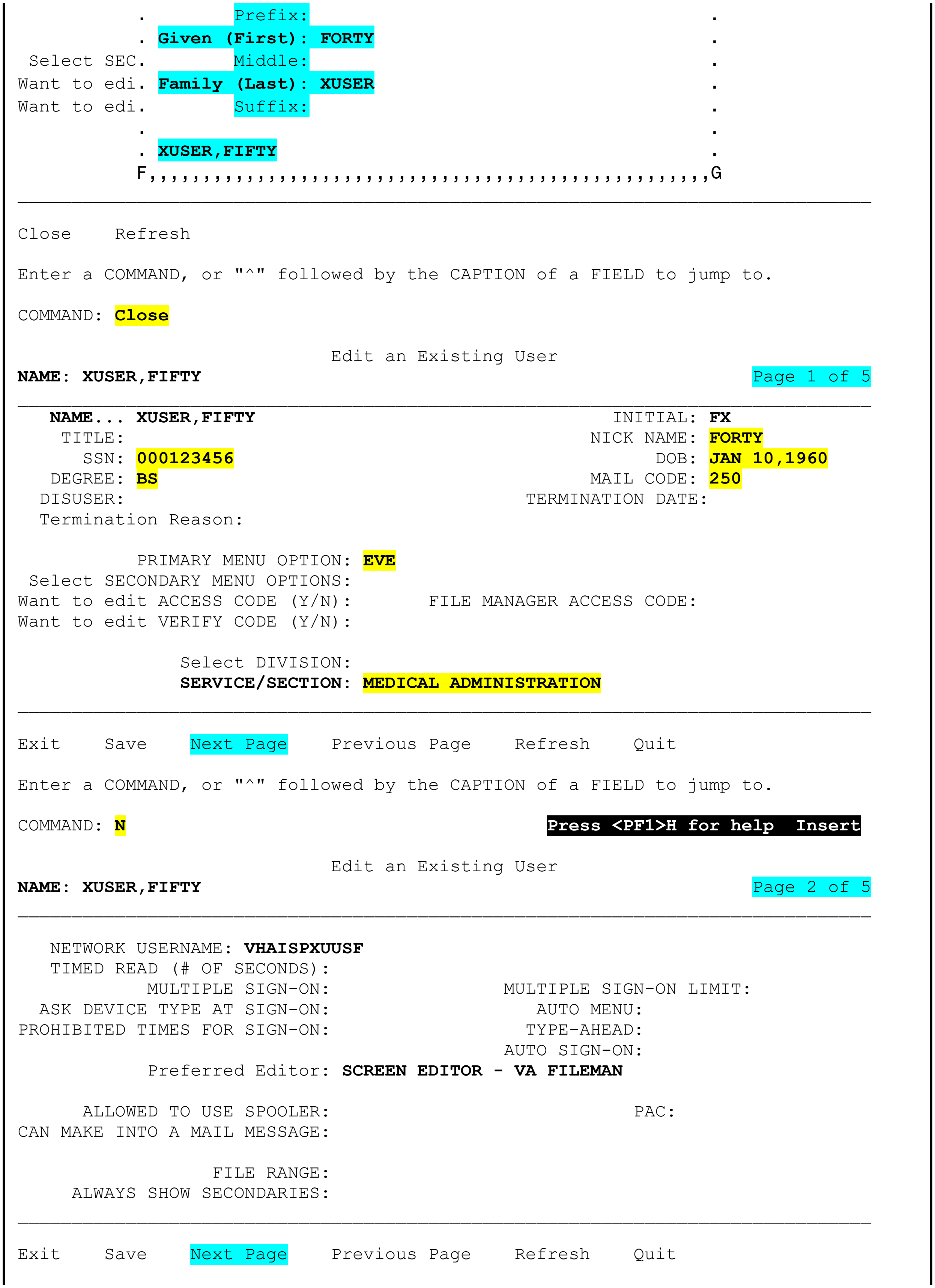
You can also print security forms for the new user with this option.

When signing on for the first time, the new user should simply press **<Enter>** at the “Verify code” prompt, which then lets them enter their own secret Verify code.

Figure 17: Add a New User to the System Legacy Option—System Prompts and User Entries and ScreenMan Forms 1 - 5

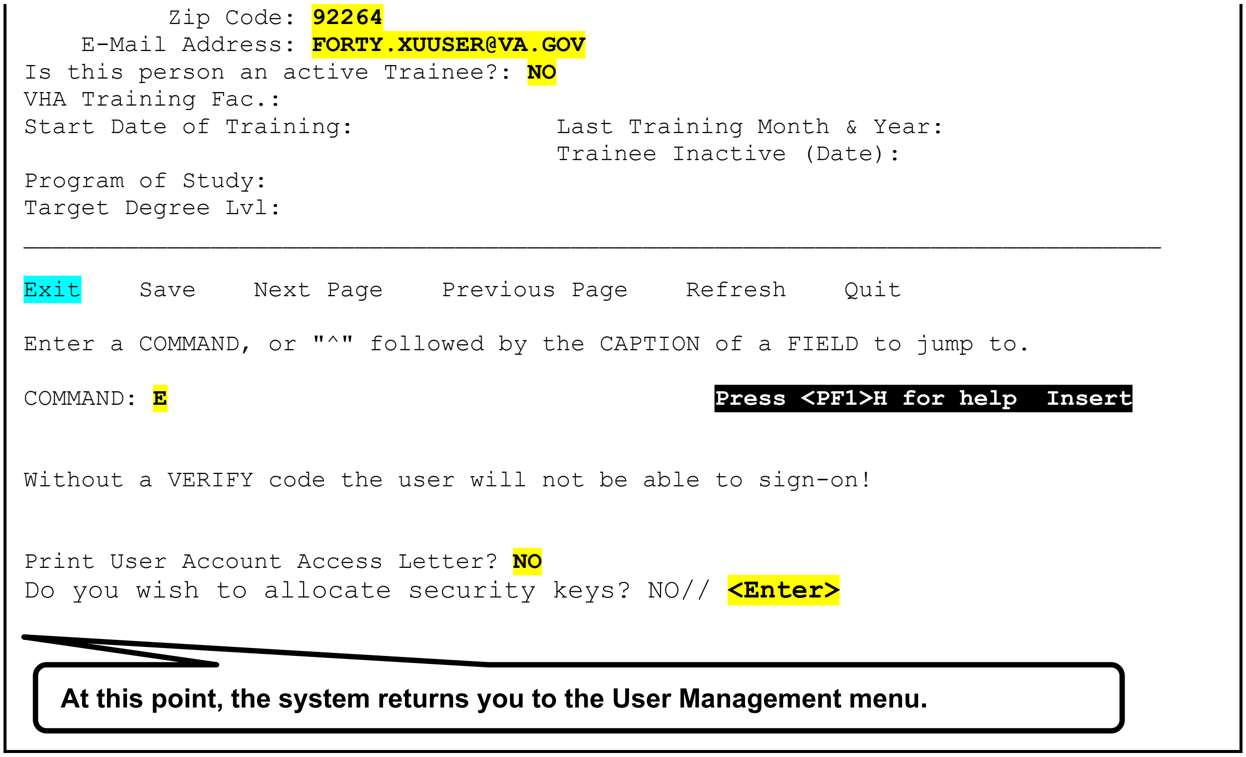


```
          Core Applications ...
          Device Management ...
   FM     VA FileMan ...
          Menu Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
          Application Utilities ...
          Capacity Planning ...
          Manage Mailman ...

Select Systems Manager Menu <TEST ACCOUNT> Option: USER <Enter> Management


          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
          Electronic Signature Block Edit
          List Inactive Person Class Users
          Manage User File ...
          OAA Trainee Registration Menu ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management <TEST ACCOUNT> Option: ADD <Enter> a New User to the System

Enter NEW PERSON's name (Family,Given Middle Suffix): XUSER,FORTY
  Are you adding 'XUSER,FIFTY' as a new NEW PERSON (the 1556TH)? No// Y <Enter>
(Yes)
Checking SOUNDEX for matches.
     USER,LAB
     USER,OPC
     USER,MC
     USER,DOCTOR
     USER,NEW
Do you still want to add this entry: NO// YES
Now for the Identifiers.
INITIAL: FX
SSN: 000123456
SEX: M <Enter> MALE
NPI: <Enter>
                             Edit an Existing User
NAME: XUSER,FORTY                                                   Page 1 of 5
_______________________________________________________________________________
   NAME... XUSER,FORTY                                 INITIAL: FX
    TITLE:                                           NICK NAME:
      SSN: 000123456                                       DOB:
   DEGREE:                                           MAIL CODE:
  DISUSER: R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
  Terminati.                                    NAME COMPONENTS .
```



```
           .        Prefix:                                     .
           . Given (First): FORTY                               .
 Select SEC.        Middle:                                     .
Want to edi. Family (Last): XUSER                               .
Want to edi.        Suffix:                                     .
           .                                                    .
           . XUSER,FIFTY                                        .
           F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
_______________________________________________________________________________

Close    Refresh

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: Close

                             Edit an Existing User
NAME: XUSER,FIFTY                                                   Page 1 of 5
_______________________________________________________________________________
   NAME... XUSER,FIFTY                                 INITIAL: FX
    TITLE:                                           NICK NAME: FORTY
      SSN: 000123456                                       DOB: JAN 10,1960
   DEGREE: BS                                        MAIL CODE: 250
  DISUSER:                                     TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS:
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE:
Want to edit VERIFY CODE (Y/N):

               Select DIVISION:
               SERVICE/SECTION: MEDICAL ADMINISTRATION
_______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: N                                       Press <PF1>H for help  Insert

                             Edit an Existing User
NAME: XUSER,FIFTY                                                   Page 2 of 5
_______________________________________________________________________________

   NETWORK USERNAME: VHAISPXUUSF
   TIMED READ (# OF SECONDS):
            MULTIPLE SIGN-ON:                MULTIPLE SIGN-ON LIMIT:
  ASK DEVICE TYPE AT SIGN-ON:                   AUTO MENU:
PROHIBITED TIMES FOR SIGN-ON:                  TYPE-AHEAD:
                                             AUTO SIGN-ON:
            Preferred Editor: SCREEN EDITOR - VA FILEMAN

      ALLOWED TO USE SPOOLER:                             PAC:
CAN MAKE INTO A MAIL MESSAGE:

                  FILE RANGE:
     ALWAYS SHOW SECONDARIES:
_______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit
```

```
Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: N                                      Press <PF1>H for help  Insert

                            Edit an Existing User
NAME: XUSER,FIFTY                                                 Page 3 of 5
______________________________________________________________________________
PROHIBITED TIMES FOR SIGN-ON:

           PHONE:                       OFFICE PHONE: 5555555555
COMMERCIAL PHONE:                         FAX NUMBER:
     VOICE PAGER:                      DIGITAL PAGER:
        LANGUAGE: ENGLISH

 Person Class                                           Effective     Expired






______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: N                                      Press <PF1>H for help  Insert

                            Edit an Existing User
NAME: XUSER,FIFTY                                                 Page 4 of 5
______________________________________________________________________________
RESTRICT PATIENT SELECTION:        OE/RR LIST:

CPRS TAB ACCESS:
  Name  Description                                 Effective Date  Expiration Date










______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: N                                      Press <PF1>H for help  Insert

                            Edit an Existing User
NAME: XUSER,FIFTY                                                 Page 5 of 5
______________________________________________________________________________
PERMANENT ADDRESS:
          Street 1: 1234 Main Street
          Street 2:
          Street 3:
              City: ANYTOWN
             State: CALIFORNIA
```



```
          Zip Code: 92264
    E-Mail Address: FORTY.XUUSER@VA.GOV
Is this person an active Trainee?: NO
VHA Training Fac.:
Start Date of Training:              Last Training Month & Year:
                                     Trainee Inactive (Date):
Program of Study:
Target Degree Lvl:
_______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: E                                      Press <PF1>H for help  Insert


Without a VERIFY code the user will not be able to sign-on!


Print User Account Access Letter? NO
Do you wish to allocate security keys? NO// <Enter>
```

At this point, the system returns you to the User Management menu.

3.2.1.1 NEW PERSON (#200) File Required Fields

When adding new users, a default set of fields is required, at a minimum. This set is defined by the NEW PERSON IDENTIFIERS field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. If it is **NULL**, the default set of required fields for the NEW PERSON (#200) file entries is:

- INITIAL (#1)
- SEX (#4)
- SSN (#9)

If, given local site policy, a different set should be used, system administrators can use this field to specify other identifiers.

NOTE: SSN is *not* required if the person entering accounts holds the XUSPF200 security key.

3.2.2 Grant Access by Profile Option

The **Grant Access by Profile** [XUSERBLK] option includes features unavailable in the **Add a New User to the System** option. With the **Grant Access by Profile** [XUSERBLK] option you can grant access to one or more people based on a typical user profile. All characteristics of the typical user, including menus, keys, and service/section, are copied to the new user or replace the characteristics of an existing user. For new users, access security forms are generated as part of the process. These forms can be delivered to the service/section coordinator by inter-office mail and can be distributed to the new users.

The **Grant Access by Profile** [XUSERBLK] option is locked with the XUMGR security key and is strictly limited for use by system administrators. It *must* be restricted, because any user profile, even that of a developer, can be copied to another user. As with the **Add a New User to the System** [XUSERNEW] option, the SSN (#9) field is required when adding new records except by holders of the XUSPF200 security key or if another default set of New Person Identifiers has been defined.

Access is assigned according to an existing user profile. Characteristics of the new user are cloned from the existing one. Rather than copying the characteristics from an actual user, creating several dummy users with profiles of typical positions can be worthwhile. A user (e.g., PHARMACY,TECH or RESIDENT,SURGERY) could be created with the appropriate user attributes, including menu options, keys, and service/section codes.

Several steps are involved in copying access to new or existing users. First you enter the name of the user account to clone from. Then, optionally, you can specify a TERMINATION DATE. Next, you enter the names of the new users to create. The system pauses for each new user as it verifies identifiers, checks for duplicates, and updates the NEW PERSON (#200) file. You *must* enter a device upon which to print the computer account notification letters. You can either run the access assignment immediately or queue it for a later time.

Figure 18: Grant Access by Profile [XUSERBLK] Legacy Option—System Prompts and User Entries

```
          Core Applications ...
          Device Management ...
   FM     VA FileMan ...
          Menu Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
          Application Utilities ...
          Capacity Planning ...
          Manage Mailman ...

Select Systems Manager Menu <TEST ACCOUNT> Option: USER <Enter> Management

          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
          Electronic Signature Block Edit
          List Inactive Person Class Users
          Manage User File ...
          OAA Trainee Registration Menu ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management <TEST ACCOUNT> Option: GRANT <Enter> Access by Profile


                          Batch Entry of New Persons
                          --------------------------

Please select a person to copy from
Template PERSON: XUSER,ONE <Enter>             OX
NAME: XUSER,ONE                         INITIAL: OX
  ACCESS CODE: <Hidden>                 FILE MANAGER ACCESS CODE: @
  DATE VERIFY CODE LAST CHANGED: FEB 10,2022
  VERIFY CODE: <Hidden>                 NICK NAME: One
  SEX: MALE
  PREFERRED EDITOR: SCREEN EDITOR - VA FILEMAN
  DATE ENTERED: DEC 08, 2021            CREATOR: XUSER,TWO
  SSN: 000123456
  LAST SIGN-ON DATE/TIME: OCT 26, 2022@15:19:34
  XUS Logon Attempt Count: 0            XUS Active User: Yes
  Entry Last Edit Date: DEC 08, 2021    TERMINAL TYPE LAST USED: C-VT320
  NAME COMPONENTS: 200
  SERVICE/SECTION: INFORMATION SYSTEMS CENTER
  SIGNATURE BLOCK PRINTED NAME: ONE XUUSER
KEY: XUPROG                             GIVEN BY: XUSER,TWO
  DATE GIVEN: DEC 08, 2021
KEY: XUMGR                              GIVEN BY: XUSER,TWO
  DATE GIVEN: DEC 08, 2021
KEY: XUPROGMODE                         GIVEN BY: XUSER,TWO
```

```
Type <Enter> to continue or '^' to exit:
  DATE GIVEN: DEC 08, 2021
KEY: XMMGR                                  GIVEN BY: XUSER,TWO
  DATE GIVEN: DEC 08, 2021
  MULTIPLE SIGN-ON: ALLOWED                 ASK DEVICE TYPE AT SIGN-ON: ASK
  AUTO MENU: YES, MENUS GENERATED           TYPE-AHEAD: ALLOWED
  TIMED READ (# OF SECONDS): 999            AUTO SIGN-ON: Yes
  PRIMARY MENU OPTION: EVE

Is this the person whose data you want cloned? Y <Enter> YES

You may enter a date, when the users that are being created/updated
will no longer have access to the system.
Enter (optional) TERMINATION DATE: <Enter>

                          Batch Entry of New Persons
                          --------------------------

Clone of: XUSER,ONE
Enter NEW PERSON's name (Family,Given Middle Suffix): XUSER,EIGHTY
  Are you adding 'XUSER,EIGHTY' as a new NEW PERSON (the 1557TH)? No// Y <Enter>
(Yes)
Checking SOUNDEX for matches.
     XUSER,FIFTY
     USER,LAB
     USER,OPC
     USER,MC
     USER,DOCTOR
     USER,NEW
Do you still want to add this entry: NO//Y
Name components.
FAMILY (LAST) NAME: XUSER// <Enter>
GIVEN (FIRST) NAME: EIGHTY// <Enter>
MIDDLE NAME: <Enter>
SUFFIX: <Enter>
Now for the Identifiers.
INITIAL: EX
SSN: 000456789
SEX: F <Enter> FEMALE
NPI:
Do You Want To Clone PERSON CLASS? YES

Next!
Enter NEW PERSON's name (Family,Given Middle Suffix): <Enter>

Where do you want to print the COMPUTER ACCOUNT NOTIFICATION LETTERS?
DEVICE: HOME// <Enter> TELNET PORT    Right Margin: 80// <Enter>


        CREATING A NEW ACCOUNT FOR 'XUSER,EIGHTY'

One moment please...
                                USER ACCOUNT NOTIFICATION

                              Department of Veterans Affairs
                                      SuperStar VAMC
                                       123 anywhere
                                   anytown, state, zip


          EIGHTY XUSER
          INFORMATION SYSTEMS CENTER   ()
```

```
         ---

         A user account has been created in your name to enable you
         to access on-line clinical and/or administrative data
         required to perform your duties as an employee of the
         Department of Veterans Affairs.  Please read the enclosed
         NEW USER INFORMATION before you attempt your first log-on
         to the system.  Questions about access should be referred
         to the AIS Application Coordinator in your service, your
         facility Information Security Officer (ISO), or your IRM
         Service.


         Your Computer Access Coordinator is:



         Your Facility Information Security Officer:

         Your Alternate Information Security Officer:





         ---


                                            Access Code: XXXXXXXX
                                            Verify Code: YYYYYYYY




                           COMPUTER ACCOUNT ACCESS POLICY

                           Department of Veterans Affairs
                                  Your VA Facility


    EIGHTY XUSER
    INFORMATION SYSTEMS CENTER   ()


As an authorized user of VHA automated information systems (AISs) and
having access to data stored in them, I will be given sufficient access to
perform my assigned duties.  I will use this access ONLY for its intended
purpose and understand the following policies that apply to VA data and
computer systems:

I agree to safeguard all passwords (e.g., Access/Verify codes, electronic
signature codes) assigned to me and am strictly prohibited from disclosing
these codes to anyone including family, friends, fellow workers,
supervisor(s), and subordinates for ANY reason.



I understand that I may be held accountable for all entries/changes made to
any government AIS using my passwords.

I am aware of the regulations and facility AIS security policies designed
```

```
 to ensure the confidentiality of all sensitive information.  I am aware
 that information about patients or employees is confidential and protected
 from unauthorized disclosure by law.  I understand that my obligation to
 protect VA information does not end with either the termination of my
 access to this facility's systems or with the termination of my government
 employment.

 I will exercise common sense and good judgment in the use of electronic
 mail.  I understand that electronic mail is not inherently confidential and
 I have no expectation of privacy in using it.  I understand that technical
 or administrative problems may create situations which requires viewing of
 my messages.  I also understand that facility management officials may
 authorize access to my electronic mail messages whenever there is a
 legitimate purpose for such access.

 I understand that a violation of this notice constitutes disregard of a
 local and/or VHA policy and will result in appropriate disciplinary action
 as defined in VA employee conduct Regulations (VAR 820(b)) as well as
 suspension/termination of access privileges.

 I affirm with my signature that I have read, understand, and agree to
 fulfill the provisions of this User Access notice.


     Signature:________________________
               EIGHTY XUSER INFORMATION SYSTEMS CENTER
     RETURN THIS FORM TO: IRMS - NEW ACCTS (xxx/xxx)


          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
          Electronic Signature Block Edit
          List Inactive Person Class Users
          Manage User File ...
          OAA Trainee Registration Menu ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management <TEST ACCOUNT> Option:
```

3.2.3 Security Forms

Figure 19: Reprint Access agreement letter Option

```
SYSTEMS MANAGER MENU ...                                                       [EVE]
User Management ...                                                          [XUSER]
   Reprint Access agreement letter                                    [XUSERREPRINT]
```

Two security forms are printed for each new user:

- **The Computer Account Notification—**Includes the user's auto-generated Access code and the name of the service/section coordinator who can answer questions.
- **The Computer Access Policy—**A contract to which users *must* adhere. It states the terms of granting access to sensitive information; the user *must* accept these terms as a condition of being given system access.

These security forms are stored in the XUSER COMPUTER ACCOUNT help frame and should be edited for local use as follows:

1. Copy the XUSER COMPUTER ACCOUNT help frame into a new site help frame (e.g., SFO COMPUTER ACCOUNT).
2. Edit the security forms for local use. Replace the "placeholder" text with the actual name and address of the facility.
3. Repoint the Kernel Parameter to the new site XUSER COMPUTER ACCOUNT help frame using VA FileMan.

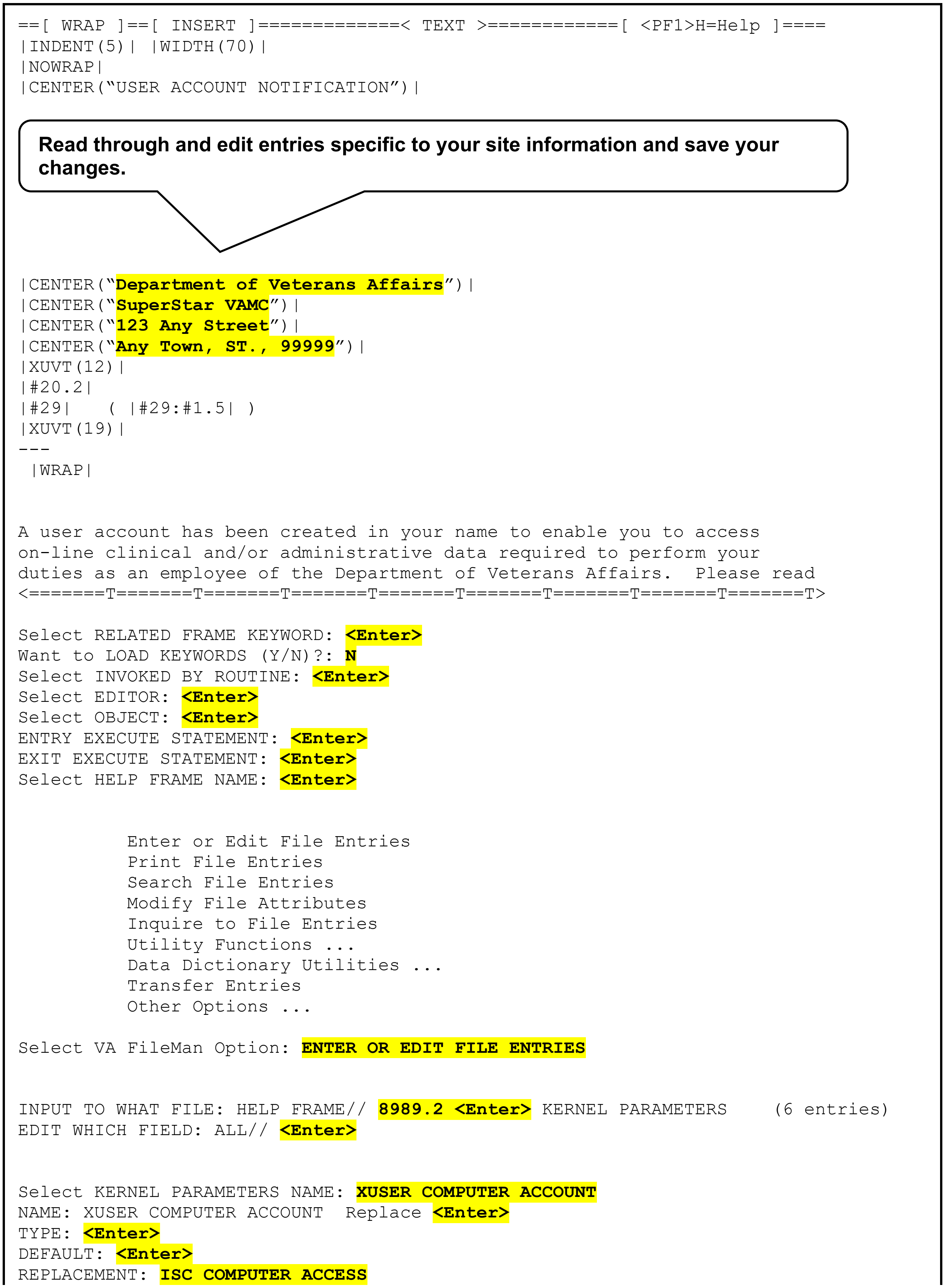
For example:

Figure 20: Security Forms—Sample User Entries (1 of 4)

```
>D ^XUP

Setting up programmer environment
This is a TEST account.

Terminal Type set to: C-VT320

You have 13 new messages.
Select OPTION NAME: SYSTEMS MANAGER MENU

          Device Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
          Application Utilities ...
          Capacity Management ...
          Manage Mailman ...
          Menu Management ...
          VA FileMan ...
          Verifier Tools Menu ...

Select Systems Manager Menu Option: VA FILEMAN

          VA FileMan Version 22.0

          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 Option: TRANSFER ENTRIES

Select TRANSFER OPTION: TRANSFER FILE ENTRIES

INPUT TO WHAT FILE: HELP FRAME// HELP FRAME <Enter>    (562 entries)
TRANSFER FROM FILE: HELP FRAME// <Enter>
TRANSFER DATA INTO WHICH HELP FRAME: ISC COMPUTER ACCESS
Not a known package or a local namespace.
  Are you adding 'ISC COMPUTER ACCESS' as a new HELP FRAME (the 563RD)? No// Y
<Enter> (Yes)
   HELP FRAME NUMBER: 742// <Enter>
   HELP FRAME HEADER: Computer Access
TRANSFER FROM HELP FRAME: XUSER COMPUTER ACCOUNT <Enter> Batch user access document
   WANT TO DELETE THIS ENTRY AFTER IT'S TRANSFERRED? No// <Enter> (No)
...SORRY, LET ME THINK ABOUT THAT A MOMENT...
SINCE THE TRANSFERRED ENTRY MAY HAVE BEEN 'POINTED TO'
BY ENTRIES IN THE 'HELP FRAME' FILE, ETC.,
DO YOU WANT THOSE POINTERS UPDATED (WHICH COULD TAKE QUITE A WHILE)? No// <Enter>
(No)
```

```
          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 Option: ENTER OR EDIT FILE ENTRIES

INPUT TO WHAT FILE: HELP FRAME// <Enter>
EDIT WHICH FIELD: ALL// TEXT <Enter>   (word-processing)

Select HELP FRAME NAME: ISC COMPUTER ACCESS <Enter>     Computer Access
NAME: ISC COMPUTER ACCESS// <Enter>
HEADER: Computer Access// <Enter>
TEXT:  . . .
       . . .
suspension/termination of access privileges.

I affirm with my signature that I have read, understand, and agree to
fulfill the provisions of this User Access notice.

|INDENT(5)||WIDTH(75)||NOWRAP|
Signature:________________________
          |#20.2| |#29|
RETURN THIS FORM TO: IRMS - NEW ACCTS (xxx/xxx)

  Edit? NO// YES
```

Figure 21: Security Forms—Sample User Entries (2 of 4)

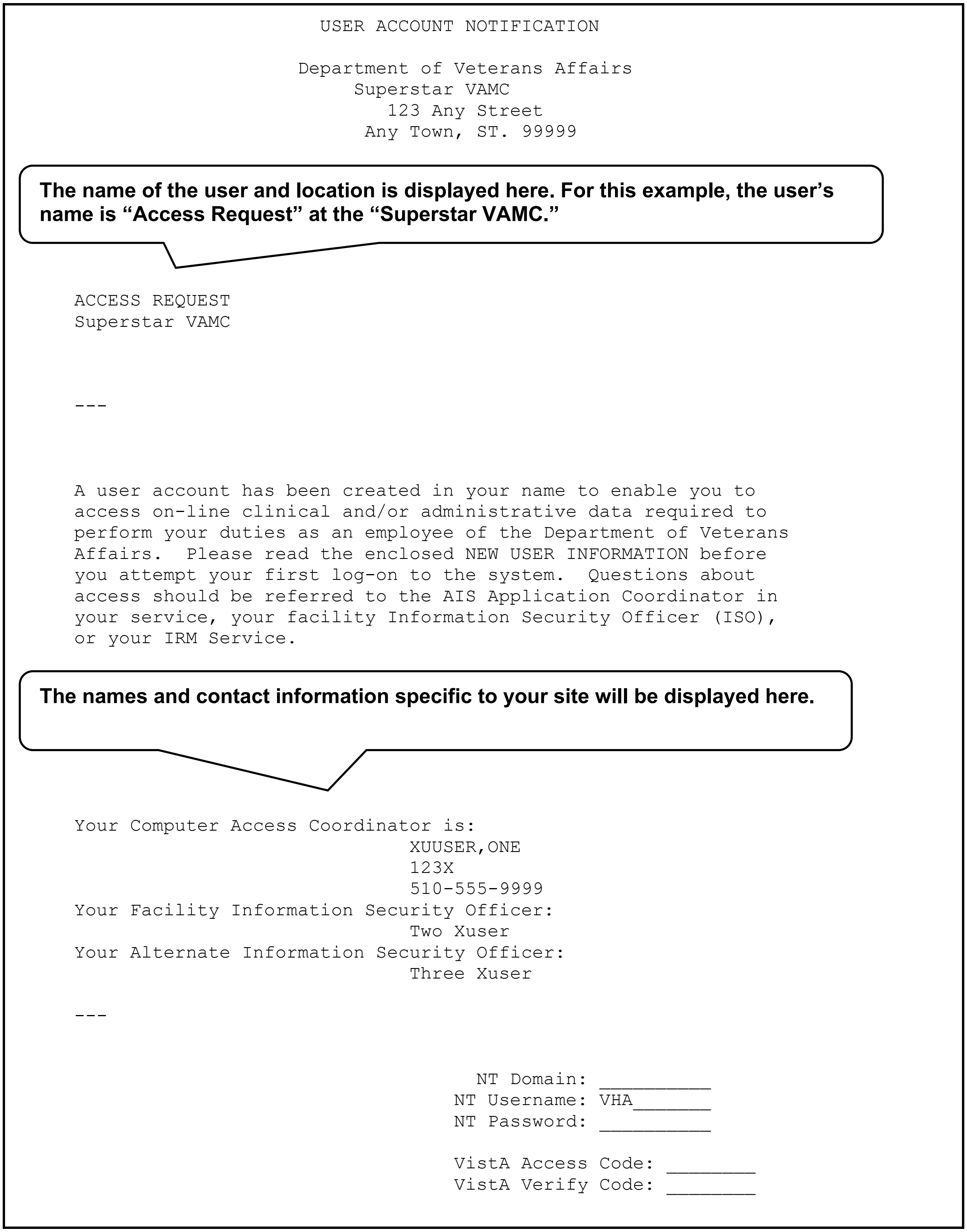


Read through and edit entries specific to your site information and save your changes.

```
==[ WRAP ]==[ INSERT ]=============< TEXT >============[ <PF1>H=Help ]====
|INDENT(5)| |WIDTH(70)|
|NOWRAP|
|CENTER("USER ACCOUNT NOTIFICATION")|


|CENTER("Department of Veterans Affairs")|
|CENTER("SuperStar VAMC")|
|CENTER("123 Any Street")|
|CENTER("Any Town, ST., 99999")|
|XUVT(12)|
|#20.2|
|#29|    ( |#29:#1.5| )
|XUVT(19)|
---
 |WRAP|


A user account has been created in your name to enable you to access
on-line clinical and/or administrative data required to perform your
duties as an employee of the Department of Veterans Affairs.  Please read
<=======T=======T=======T=======T=======T=======T=======T=======T=======T>

Select RELATED FRAME KEYWORD: <Enter>
Want to LOAD KEYWORDS (Y/N)?: N
Select INVOKED BY ROUTINE: <Enter>
Select EDITOR: <Enter>
Select OBJECT: <Enter>
ENTRY EXECUTE STATEMENT: <Enter>
EXIT EXECUTE STATEMENT: <Enter>
Select HELP FRAME NAME: <Enter>


          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 Option: ENTER OR EDIT FILE ENTRIES


INPUT TO WHAT FILE: HELP FRAME// 8989.2 <Enter> KERNEL PARAMETERS    (6 entries)
EDIT WHICH FIELD: ALL// <Enter>


Select KERNEL PARAMETERS NAME: XUSER COMPUTER ACCOUNT
NAME: XUSER COMPUTER ACCOUNT  Replace <Enter>
TYPE: <Enter>
DEFAULT: <Enter>
REPLACEMENT: ISC COMPUTER ACCESS
```

```
Select KERNEL PARAMETERS NAME: <Enter>


          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 Option: <Enter>


   FM     VA FileMan ...
          Core Applications ...
          Device Management ...
          Information Security Officer Menu ...
          Manage Mailman ...
          Menu Management ...
          Operations Management ...
          Programmer Options ...
          Spool Management ...
          Taskman Management ...
          User Management ...

Select Systems Manager Menu Option: USER MANAGEMENT

          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
          Electronic Signature Block Edit
          Manage User File ...
          OAA Trainee Registration Menu ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management Option: REPRINT ACCESS AGREEMENT LETTER
Select NEW PERSON NAME: REQUEST,ACCESS <Enter>     AR     COMPUTER SPECIALIST

 Is REQUEST,ACCESS the one you want? YES// <Enter>
DEVICE: 0;80;60 <Enter>  Telnet Terminal
```

Figure 22: Security Forms—Sample User Account Notification Form (3 of 4)



```
                      USER ACCOUNT NOTIFICATION

                    Department of Veterans Affairs
                         Superstar VAMC
                            123 Any Street
                          Any Town, ST. 99999
```

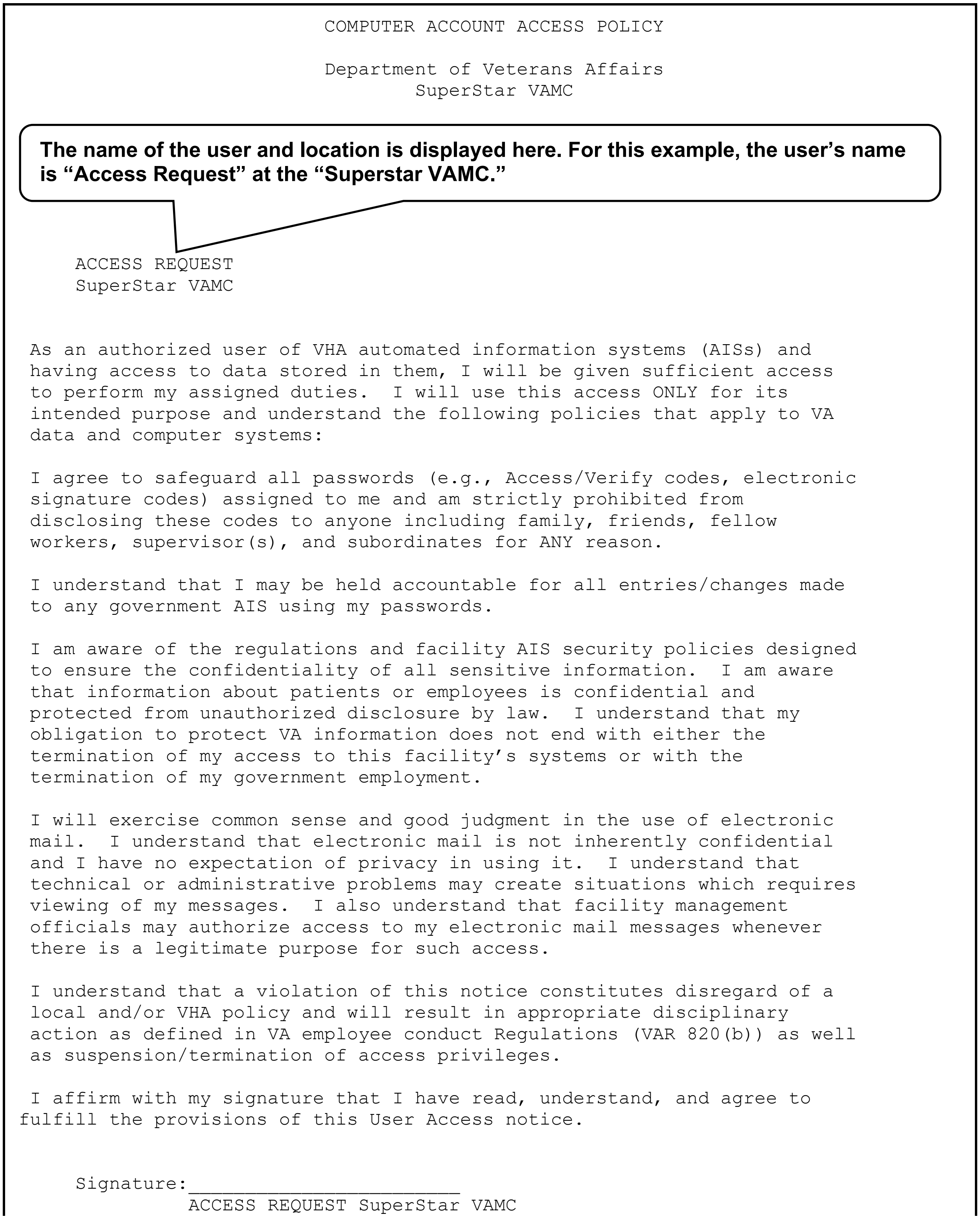
The name of the user and location is displayed here. For this example, the user's name is "Access Request" at the "Superstar VAMC."

```
ACCESS REQUEST
Superstar VAMC



---



A user account has been created in your name to enable you to
access on-line clinical and/or administrative data required to
perform your duties as an employee of the Department of Veterans
Affairs.  Please read the enclosed NEW USER INFORMATION before
you attempt your first log-on to the system.  Questions about
access should be referred to the AIS Application Coordinator in
your service, your facility Information Security Officer (ISO),
or your IRM Service.
```

The names and contact information specific to your site will be displayed here.

```
Your Computer Access Coordinator is:
                              XUUSER,ONE
                              123X
                              510-555-9999
Your Facility Information Security Officer:
                              Two Xuser
Your Alternate Information Security Officer:
                              Three Xuser

---


                                    NT Domain: __________
                                  NT Username: VHA_______
                                  NT Password: __________

                                  VistA Access Code: ________
                                  VistA Verify Code: ________
```

Figure 23: Security Forms—Sample Computer Account Access Policy Form (4 of 4)



The name of the user and location is displayed here. For this example, the user's name is "Access Request" at the "Superstar VAMC."

```
                            COMPUTER ACCOUNT ACCESS POLICY

                            Department of Veterans Affairs
                                    SuperStar VAMC


     ACCESS REQUEST
     SuperStar VAMC


 As an authorized user of VHA automated information systems (AISs) and
 having access to data stored in them, I will be given sufficient access
 to perform my assigned duties.  I will use this access ONLY for its
 intended purpose and understand the following policies that apply to VA
 data and computer systems:

 I agree to safeguard all passwords (e.g., Access/Verify codes, electronic
 signature codes) assigned to me and am strictly prohibited from
 disclosing these codes to anyone including family, friends, fellow
 workers, supervisor(s), and subordinates for ANY reason.

 I understand that I may be held accountable for all entries/changes made
 to any government AIS using my passwords.

 I am aware of the regulations and facility AIS security policies designed
 to ensure the confidentiality of all sensitive information.  I am aware
 that information about patients or employees is confidential and
 protected from unauthorized disclosure by law.  I understand that my
 obligation to protect VA information does not end with either the
 termination of my access to this facility's systems or with the
 termination of my government employment.

 I will exercise common sense and good judgment in the use of electronic
 mail.  I understand that electronic mail is not inherently confidential
 and I have no expectation of privacy in using it.  I understand that
 technical or administrative problems may create situations which requires
 viewing of my messages.  I also understand that facility management
 officials may authorize access to my electronic mail messages whenever
 there is a legitimate purpose for such access.

 I understand that a violation of this notice constitutes disregard of a
 local and/or VHA policy and will result in appropriate disciplinary
 action as defined in VA employee conduct Regulations (VAR 820(b)) as well
 as suspension/termination of access privileges.

 I affirm with my signature that I have read, understand, and agree to
fulfill the provisions of this User Access notice.


     Signature:________________________
               ACCESS REQUEST SuperStar VAMC
     RETURN THIS FORM TO: IRMS - NEW ACCTS (xxx/xxx)
```

The name of the user and location is displayed here.

VA FileMan word-processing “windows” are used to retrieve the:

- User’s name
- Service/Section
- Service/Section coordinator’s name

To be effective, the SERVICE/SECTION field in the NEW PERSON (#200) file *must* be filled in for the new user. The COORDINATOR (IRM) field, a field in the SERVICE/SECTION (#49) file, *must* also be filled in and updated when necessary. Word-processing “windows” are also used for formatting, like |TOP|, to separate the two forms. When using the File Access Security system, **READ** access to the SERVICE/SECTION (#49) file is needed to retrieve the Coordinator’s name within the window command.

REF: For more information on using word-processing “windows,” the File Access Security system, and navigation, see the *VA FileMan User Manual*.

The **Reprint Access Agreement Letter** [XUSERREPRINT] option allows you to reprint the computer access agreement letter in case there was a problem printing the first form (e.g., the first form is jammed in the printer). It does *not* reprint the Access code on the letter, however.

3.3 Edit an Existing User Option

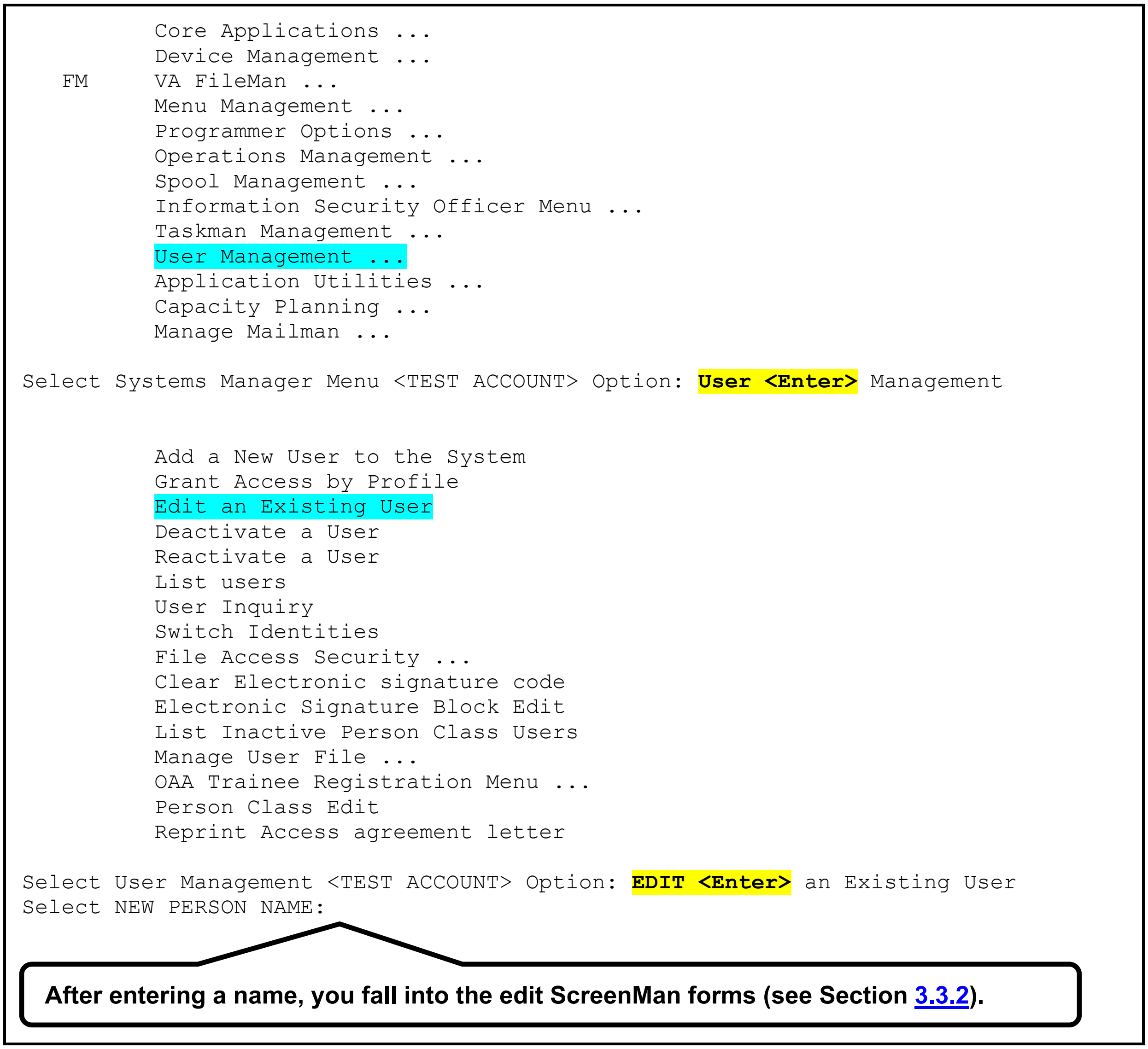
Figure 24: Edit an Existing User Option—Menu

```
SYSTEMS MANAGER MENU ...                                              [EVE]
User Management ...                                                 [XUSER]
   Edit an Existing User                                        [XUSEREDIT]
```

The attributes of an existing user can be edited with the **Edit an Existing User** [XUSEREDIT] option. This option invokes a screen-oriented display using ScreenMan.

It is impossible to exit the form and save changes unless all required fields (e.g., the SERVICE/SECTION field in the NEW PERSON [#200] file) are filled in (see Section 3.3.2).

Figure 25: Edit an Existing User [XUSEREDIT] Option



```
          Core Applications ...
          Device Management ...
   FM     VA FileMan ...
          Menu Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
          Application Utilities ...
          Capacity Planning ...
          Manage Mailman ...

Select Systems Manager Menu <TEST ACCOUNT> Option: User <Enter> Management


          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
          Electronic Signature Block Edit
          List Inactive Person Class Users
          Manage User File ...
          OAA Trainee Registration Menu ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management <TEST ACCOUNT> Option: EDIT <Enter> an Existing User
Select NEW PERSON NAME:
```

After entering a name, you fall into the edit ScreenMan forms (see Section 3.3.2).

3.3.1 Editable Field Attributes

Table 4 describes each of the user field attributes you can edit with the **Edit an Existing User** [XUSEREDIT] option.

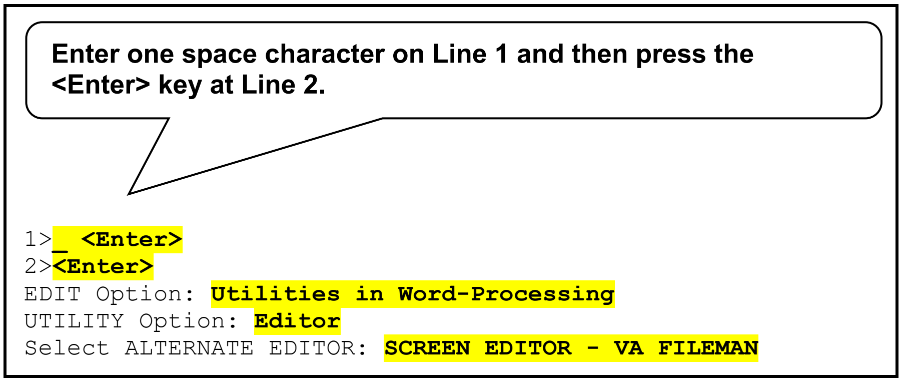
Table 4: Edit an Existing User Option—Editable Field Attributes

Field Attribute	Description
NAME (#.01) (Required)	The user's name should be entered in capital letters. The syntax should be "LAST,FIRST MI." with only a comma (no spaces) between the last and first name. A middle initial can follow, separated with a space and followed with a period. It is *not* appropriate to add credentials (e.g., M.D.), since there are other ways to specify such additional information (by the Title and the Signature Block Printed Name). Furthermore, the parsing algorithms commonly used in software applications only recognize two pieces, before and after the comma, rearranging them and using uppercase/lowercase to generate "First MI. Last".
INITIAL (#1)	The user's initials can be entered, usually two or three capital letters with no spaces. The NEW PERSON (#200) file contains a lookup-type cross-reference by INITIAL (C), so if the INITIAL field is filled in, the user can be found in the NEW PERSON (#200) file by entering the initials. For example, just the initials can be used at the "Select NEW PERSON Name:" prompt, or when addressing mail messages, or for other lookup purposes. Users can edit their initials at any time since this field is included in the common **Edit User Characteristics** [XUSEREDITSELF] option.
TITLE (#8)	This field points to the TITLE (#3.1) file, a file exported with Kernel but without data (records). The User Management options to add or edit a user's record allow **LAYGO** into the TITLE (#3.1) file, so titles can be added via the NEW PERSON (#200) file. Although *not* required, it may be wise to assign appropriate titles to users, so this field can be referenced by other software applications. MailMan, for example, displays titles in message headers if the user who is reading mail has so indicated with a flag in MailMan's Edit User Options called **Show Titles**.
NICK NAME (#13)	Like INITIAL, NICK NAME has a lookup type cross-reference (D) in the NEW PERSON (#200) file so that lookups succeed simply by using the NICK NAME. This field is also included in Edit User Characteristics.
SSN (#9)	The SSN (#9) field is *not* a required field in the data dictionary for the NEW PERSON (#200) file. SSN is required when using the User Management options to add a new user unless the XUSPF200 security key is held by the person using the option. It is *highly recommended* that each new user have the SSN (#9) field filled in to minimize the problem of subsequent duplicate entries. Since many existing users do *not* have an SSN entered, however,

Field Attribute	Description
	the **Edit an Existing User** [XUSEREDIT] option does *not* require that one be entered.
MAIL CODE (#28)	The user's MAIL CODE can be entered for purposes of interoffice routing of manually delivered mail.
PRIMARY MENU OPTION (#201) (Required for functional access)	Users *must* be assigned a PRIMARY MENU OPTION (#201) field in order to reach Menu Manager after successfully entering Access and Verify codes. The PRIMARY MENU OPTION should provide a route to all the computing functions the user can be expected to need. The XUMGR security key *must* be held by the person assigning the menu (unless delegated options are available for use with the Secure Menu Delegation system). **REF:** Building and rearranging menus is discussed in the "Menu Manager: System Management" section.
SECONDARY MENU OPTIONS (#203)	The SECONDARY MENU OPTIONS (#203) field can be used to assign particular options to individual users to customize their menu choices. While a user may have a standard primary menu to carry out the usual functions of a department or service, additional special functions just for this user can be assigned as secondary options. This is a multiple field, unlike the PRIMARY MENU OPTION (#201) field, so additional items can easily be added.
ACCESS CODE (#2) VERIFY CODE (#7.2)	These fields can be used to edit a user's Access or Verify Code as needed. If a user has forgotten the Verify code, or needs a new one, system administrators/ISO should delete the existing code so that when the user logs on and presses the <Enter> key at the "VERIFY CODE" prompt, a new (secret) password (VERIFY CODE) can be entered. To accomplish this, "Y" should be entered at the "Want to edit VERIFY CODE (Y/N) :" prompt. An at-sign (**@**) should then be entered to delete the existing code. The change is filed immediately, unlike other changes that are processed as part of the overall transaction when leaving the ScreenMan form. Users can edit their Verify code at any time via the **Edit User Characteristics** [XUEDITSELF] option on the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). If this option uses a local template, the ability to edit the VERIFY CODE field should probably remain, as a security measure. System administrators can choose to add the ability to edit the ACCESS CODE field as well. **REF:** For more information on the **Edit User Characteristics** [XUEDITSELF] option, see the "Edit User Characteristics Option" section.

Field Attribute	**Description**
FILE MANAGER ACCESS CODE (#3)	The FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON [#200] file) is stored in the local variable **DUZ(0)**. If **DUZ(0)=@**, the user is a developer with the highest level of Programmer access authority. Other *non*-reserved symbols can be assigned for File Access Security, depending on the user's needs. Software applications indicate which symbols are needed for site-specific File Access Security. **NOTE:** In previous documentation and data dictionaries, it has been *implied* that the hashtag (**#**) symbol/character was reserved for File Access Security for system administrators; however, this is *not* true. It has merely been used as a *convention.* If the File Access Security conversion has been run, the FILE MANAGER ACCESS CODE (#3) field is *not* used to control file-level access security as it was *before* the conversion. The File Access Security system (formerly known as Part 3 of the Kernel installation) permits the association of a user with a file whereby explicit access can be granted. While the conversion process is somewhat involved, the benefits resulting from implementing the File Access Security system are worthwhile. Even after running the file access conversion, the FILE MANAGER ACCESS CODE (#3) field continues to serve several functions: If a user has been granted full file access privileges for a particular file, a further restriction can be placed at the file or field level to prohibit modification of the definition or entry of data. Files have top-level restrictions of **READ**, **WRITE**, or **DELETE** access as do fields and templates. If the file, field, or template is protected with the at-sign (**@**; Programmer access), the user *must* also have the at-sign in the FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON (#200) file. The Device Handler also checks the FILE MANAGER ACCESS CODE (#3) field of the user if the SECURITY field in the DEVICE (#3.5) file has been defined with a character string. The user would *not* be able to select the device unless at least one of the characters in the user's code matched at least one character in the device code. The most important FILE MANAGER ACCESS CODE (#3) field character is the at-sign (**@**; Programmer access). It has special meaning and overrides other file access restrictions or other FILE MANAGER ACCESS CODE (#3) field characters. It is *not* recommended that the at-sign be allocated unless absolutely needed. Allocation is, in part, restricted by the fact that only those few users who have Programmer access to the system can give other

Field Attribute	Description
	users the at-sign. **NOTE:** A **SET** statement from programmer mode can be used to temporarily assign **DUZ(0)="@"** *without* storing the code in the NEW PERSON (#200) file, which would give permanent Programmer access. Use of the at-sign (**@**; Programmer access) is less common now than in the past since alternative security measures have been developed. It is still required for several critically sensitive checks, however, such as entering M code into VA FileMan files (e.g., OPTION [#19] and FUNCTION [#.5] files). **REF:** For more information on File Access Security, see "File Access Security" in this manual and the *VA FileMan (Version 22.0) and Kernel (Version 8.0) File Access Security* supplemental documentation located on the VA Software Document Library (VDL) at: VDL VA FileMan Application Documents

PREFERRED EDITOR (#31.3)	If a user's PREFERRED EDITOR field is **NULL**, Kernel uses VA FileMan's Line Editor to edit word-processing fields. If the PREFERRED EDITOR is set to another entry in the ALTERNATE EDITOR (#1.2) file, like VA FileMan's Screen Editor, Kernel uses that editor when the user edits word-processing fields. As described in VA FileMan's documentation, users can switch from the Line Editor to another editor by using the Utility suboption on the **Edit options** [XUEDITOPT] menu.
	Figure 26: VA FileMan Line Editor—Sample User Dialog
	
	Enter one space character on Line 1 and then press the <Enter> key at Line 2.
	`1>`**`_ <Enter>`**
	`2>`**`<Enter>`**
	`EDIT Option: `**`Utilities in Word-Processing`**
	`UTILITY Option: `**`Editor`**
	`Select ALTERNATE EDITOR: `**`SCREEN EDITOR - VA FILEMAN`**

Field Attribute	Description
	If the PREFERRED EDITOR is the Screen Editor, it is also possible to switch to another editor, like the Line Editor, to take advantage of Line Editor features such as File Transfer from Foreign CPU. **NOTE:** Other editors (e.g., Class III editors) may *not* support switching to the Line Editor, which may be a limitation in some circumstances. This field is also included in **Edit User Characteristics** [XUSEREDITSELF] and MailMan's Edit User options, so that all users can define a PREFERRED EDITOR if they so choose.
DIVISION (#16)	The DIVISION Multiple field has a corresponding site parameter, the Default Institution, that sets users' **DUZ(2)** if this field is *not* filled in. A user setting, however, takes precedence over the site parameter. This is a multiple field and if the user is associated with more than one institution, the user is prompted at signon to pick the one corresponding to the computing activities to be carried out in that session.
SERVICE/SECTION (#29) (Required)	This field points to the SERVICE/SECTION (#49) file distributed with Kernel's virgin installation. No data is included. It is a required field since applications have begun to use it in various utilities. Kernel's **CPU/Service/User/Device Stats** [XUSTAT] option, for example, can summarize signon information for all users in the same Service/Section. The **Grant Access by Profile** [XUSERBLK] option also makes use of this field to specify the Service/Section Coordinator to whom the access forms of the new users should be delivered.
NETWORK USERNAME (#501.1)	This is the username that is used by the Windows Active Directory. It can be used to help identify the user; although it should *not* be relied on for accuracy as it is manually entered data that is *not* validated by Active Directory.
TIMED READ (#200.1)	As discussed with other site parameters earlier in this section, TIMED READ defines the length of time Kernel should wait for a user response to a **READ**. A setting for the user attribute overrides the site default. It is used to define the local variable **DTIME**.
MULTIPLE SIGN-ON (#200.04)	As discussed with other site parameters, this field controls whether the user is permitted to have two or more concurrent signon sessions. The user setting takes precedence.
AUTO MENU (#200.06)	As discussed with other site parameters, this field controls whether the entire list of menu options is automatically presented or whether the user needs to enter a question mark (**?**) to invoke the display. The user setting takes precedence.

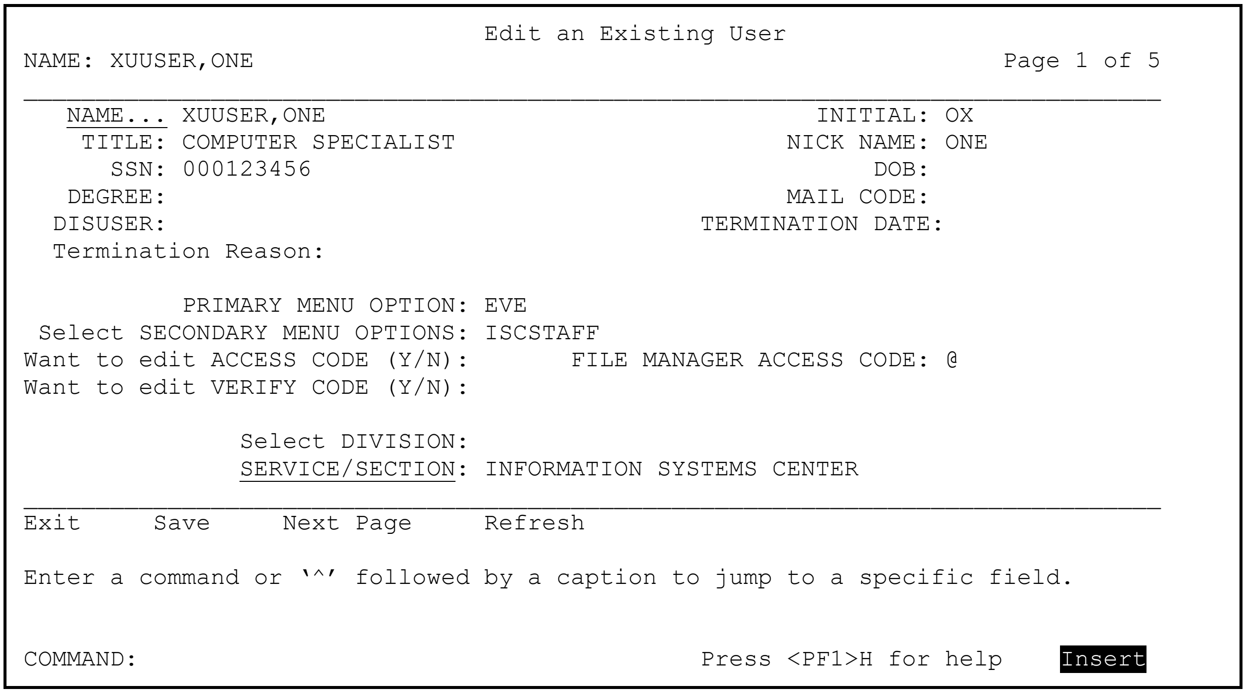
Field Attribute	Description
ASK DEVICE TYPE AT SIGN-ON (#200.05)	As discussed with other site parameters, this field controls whether the device being used at signon is queried for its terminal type. The user setting takes precedence.
TYPE-AHEAD (#200.09)	This field controls whether the user can enter text faster than the computer can read it. If set to **YES**, the computer buffers input from the user. If set to **NO**, keystrokes from the user are lost if they are typed faster than the computer can process them.
ALLOWED TO USE SPOOLER (#41)	This field controls whether a user can pick the spool device at the device prompt to send output to the spooler.
PAC (#14, Programmer Access Code)	For users who have been granted the **Programmer mode** [XUPROGMODE] option along with the XUPROG and XUPROGMODE security keys, a Programmer Access Code can be assigned as additional security. If a PAC is defined, Kernel prompts for the PAC just before allowing a user to enter programmer mode. If this field is **NULL**, a PAC is *not* asked.
CAN MAKE INTO A MAIL MESSAGE (#41.2)	This field controls whether a spooled document can be transformed into a regular mail message for use within MailMan.
DISUSER (#7)	If set to **YES**, disables access to the system for this user (without terminating the user's account).
FILE RANGE (#31.1)	Users who have VA FileMan privileges to create files can be given a numeric range of numbers to use as file numbers. Assigning number ranges acts as a safeguard to keep users from picking a number within a range that is nationally reserved for VistA software applications. It can also serve local database administration needs of segmenting local development by number ranges.
TERMINATION DATE (#9.2)	As described in the "Deactivating Users" section, this field indicates when a user's access privileges should be revoked.
ALWAYS SHOW SECONDARIES (#200.11)	If set to **YES**, contents of a user's SECONDARY MENU OPTIONS (#203) are shown when the user enters one question mark (**?**) at a menu prompt. Otherwise, the user *must* enter two question marks (**??**) to see their secondary menu.
PROHIBITED TIMES FOR SIGN-ON (#15)	As discussed with other signon parameters, this field can be used to regulate when the user can sign on to the system. The user setting takes precedence over any corresponding device setting.
PHONE (HOME) (#.131) OFFICE PHONE (#.132) PHONE #3 (#.133) PHONE #4 (#.134)	Set up phone numbers for the user in these fields.

Field Attribute	Description
COMMERCIAL PHONE (#.135) FAX NUMBER (#.136)	
VOICE PAGER (#.137) DIGITAL PAGER (#.138)	Set up pager numbers for the user in these fields.
LANGUAGE (#200.07)	Overrides the setting of the DEFAULT LANGUAGE field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. Both of these are used to set the **DUZ(“LANG”)** flag for each user. VA FileMan uses this setting to enable the display of language-specific dates and times, numeric formats, and dialogs.

3.3.2 ScreenMan Forms

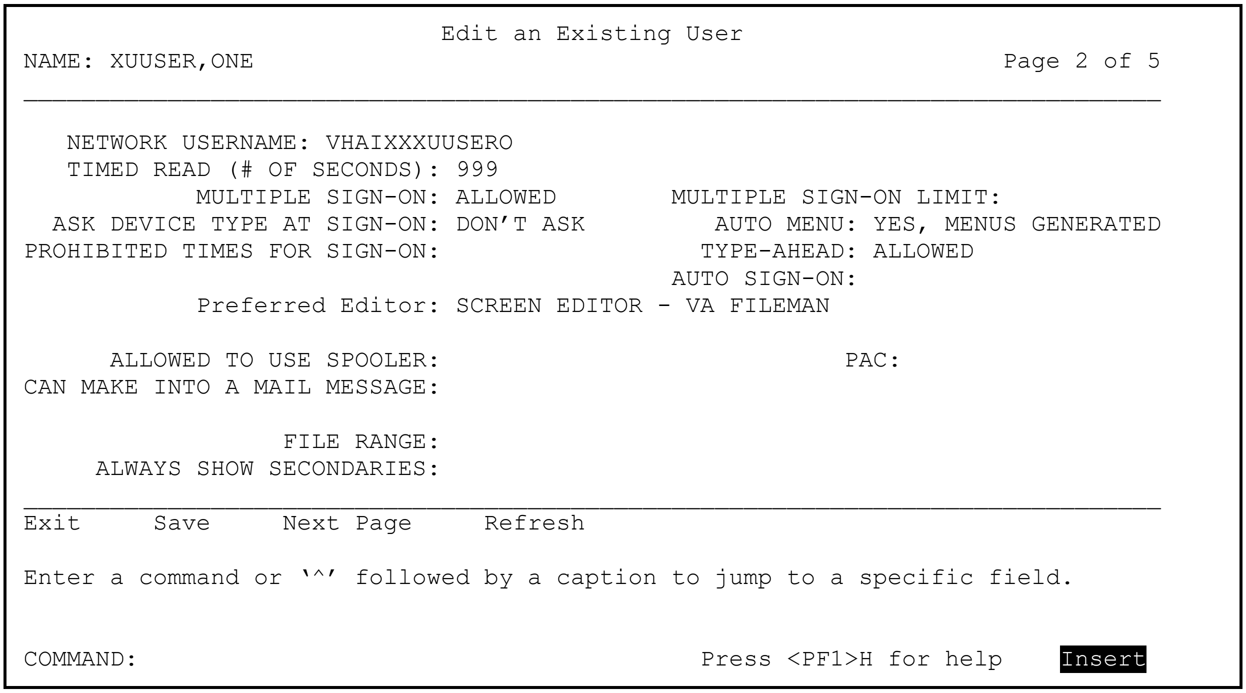
Figure 27 - Figure 31 are examples of the five ScreenMan forms displaying the fields that you can edit with the **Edit an Existing User** [XUSEREDIT] option:

Figure 27: Edit an Existing User Option—Screen 1



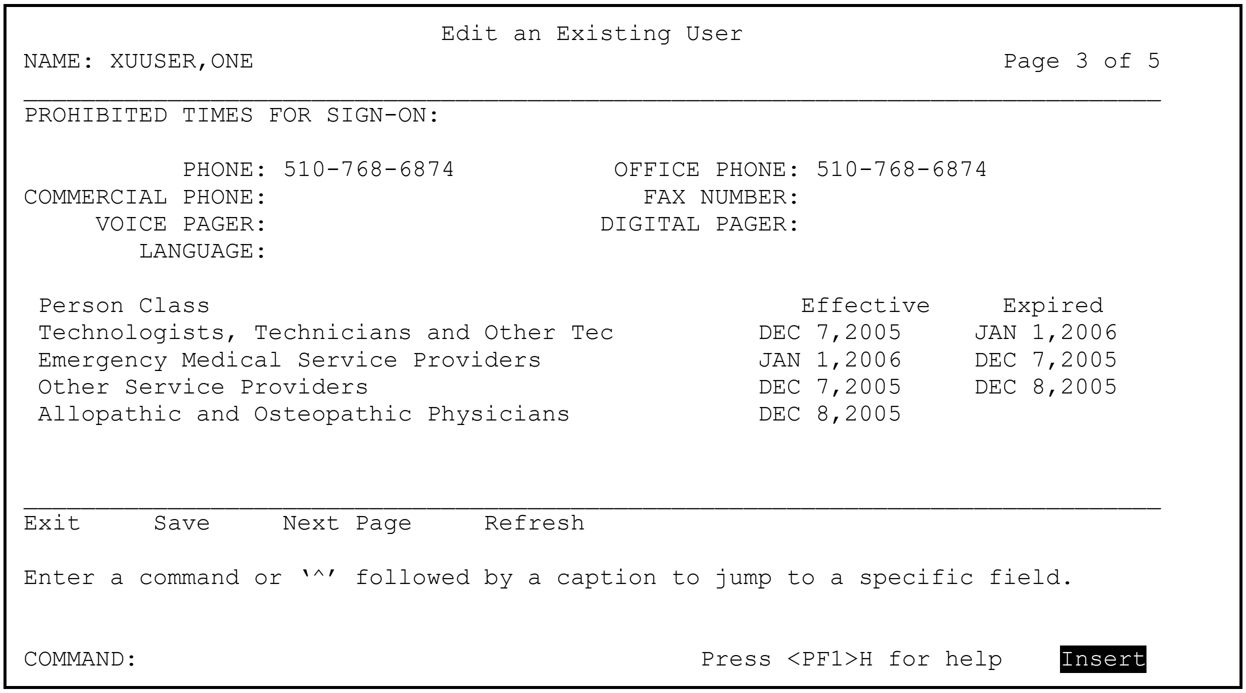
```
                                Edit an Existing User
NAME: XUUSER,ONE                                                    Page 1 of 5
_______________________________________________________________________________
   NAME... XUUSER,ONE                                  INITIAL: OX
    TITLE: COMPUTER SPECIALIST                       NICK NAME: ONE
      SSN: 000123456                                       DOB:
   DEGREE:                                           MAIL CODE:
  DISUSER:                                     TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS: ISCSTAFF
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION:
               SERVICE/SECTION: INFORMATION SYSTEMS CENTER
_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or ‘^’ followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

Figure 28: Edit an Existing User Option—Screen 2



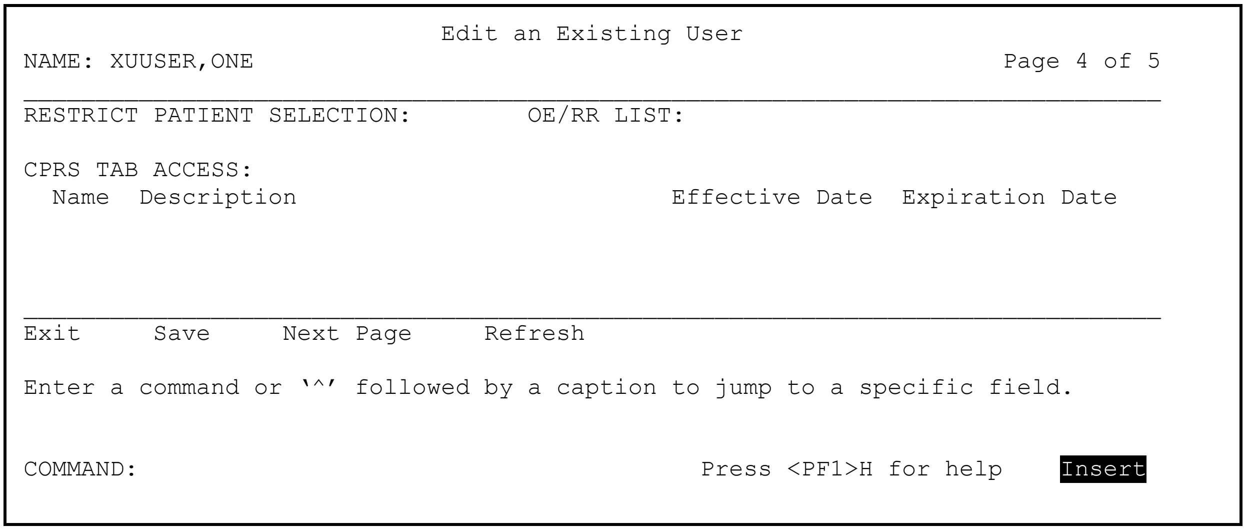
```
                             Edit an Existing User
NAME: XUUSER,ONE                                                    Page 2 of 5
_______________________________________________________________________________

   NETWORK USERNAME: VHAIXXXUUSERO
   TIMED READ (# OF SECONDS): 999
            MULTIPLE SIGN-ON: ALLOWED        MULTIPLE SIGN-ON LIMIT:
  ASK DEVICE TYPE AT SIGN-ON: DON’T ASK         AUTO MENU: YES, MENUS GENERATED
PROHIBITED TIMES FOR SIGN-ON:                  TYPE-AHEAD: ALLOWED
                                             AUTO SIGN-ON:
            Preferred Editor: SCREEN EDITOR - VA FILEMAN

      ALLOWED TO USE SPOOLER:                            PAC:
CAN MAKE INTO A MAIL MESSAGE:

                  FILE RANGE:
     ALWAYS SHOW SECONDARIES:
_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or ‘^’ followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

Figure 29: Edit an Existing User Option—Screen 3



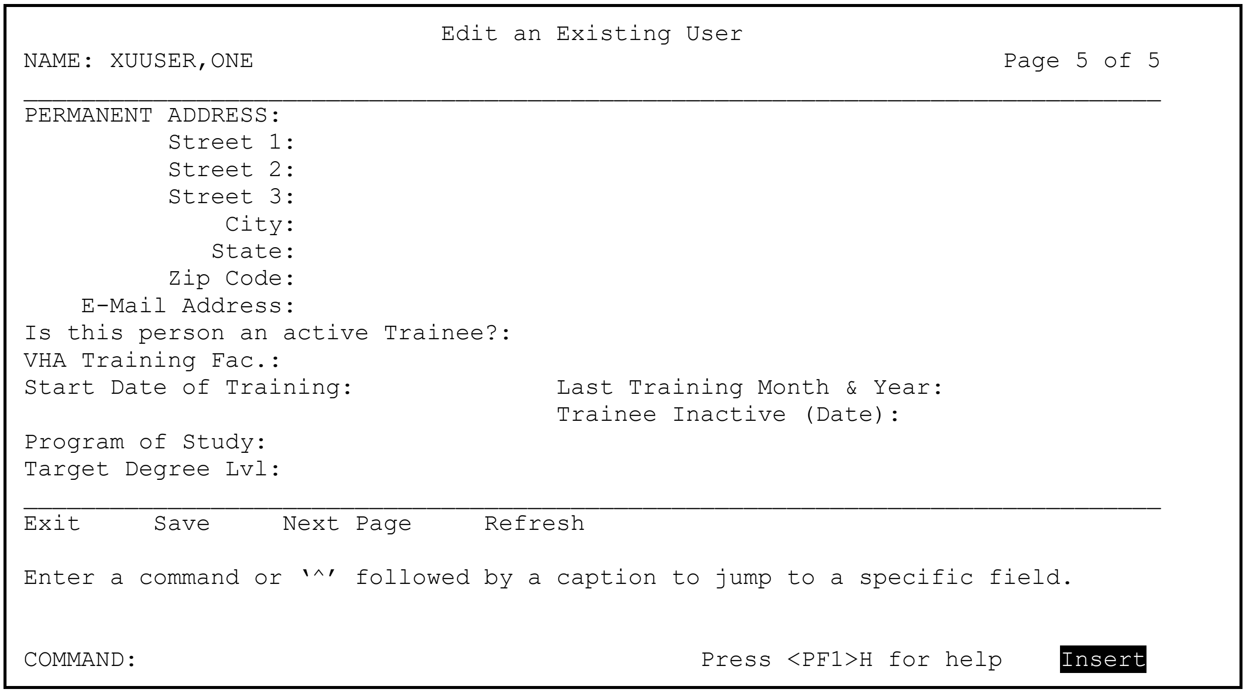
```
                             Edit an Existing User
NAME: XUUSER,ONE                                                    Page 3 of 5
_______________________________________________________________________________
PROHIBITED TIMES FOR SIGN-ON:

           PHONE: 510-768-6874           OFFICE PHONE: 510-768-6874
COMMERCIAL PHONE:                          FAX NUMBER:
     VOICE PAGER:                       DIGITAL PAGER:
        LANGUAGE:

 Person Class                                         Effective     Expired
 Technologists, Technicians and Other Tec          DEC 7,2005     JAN 1,2006
 Emergency Medical Service Providers               JAN 1,2006     DEC 7,2005
 Other Service Providers                           DEC 7,2005     DEC 8,2005
 Allopathic and Osteopathic Physicians             DEC 8,2005



_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or ‘^’ followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

Figure 30: Edit an Existing User Option—Screen 4



```
                             Edit an Existing User
NAME: XUUSER,ONE                                                    Page 4 of 5
_______________________________________________________________________________
RESTRICT PATIENT SELECTION:        OE/RR LIST:

CPRS TAB ACCESS:
  Name  Description                                  Effective Date  Expiration Date




_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

Figure 31: Edit an Existing User Option—Screen 5



```
                             Edit an Existing User
NAME: XUUSER,ONE                                                    Page 5 of 5
_______________________________________________________________________________
PERMANENT ADDRESS:
          Street 1:
          Street 2:
          Street 3:
              City:
             State:
          Zip Code:
    E-Mail Address:
Is this person an active Trainee?:
VHA Training Fac.:
Start Date of Training:              Last Training Month & Year:
                                     Trainee Inactive (Date):
Program of Study:
Target Degree Lvl:
_______________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

3.3.3 Additional Attributes Editable by Users

Some but *not* all of the user attribute fields can be edited by users using the **Edit User Characteristics** [XUSEREDITSELF] option. The only field the user can edit that is *not* part of the system manager's Edit an Existing User form is the TEXT TERMINATOR field.

REF: For a description of the fields users can edit (using the default Edit User Characteristics form and template), see Table 3 in the "Edit User Characteristics Option" section.

3.3.4 Edit User Characteristics Form and Template

Kernel exports a ScreenMan form and a template to be used in the **Edit User Characteristics** [XUSEREDITSELF] option. Both are called XUEDIT CHARACTERISTICS. The INPUT template by the same name is invoked if the ScreenMan form *cannot* be loaded on the current terminal type.

System administrators can substitute a locally-developed template by entering its name in the USER CHARACTERISTICS TEMPLATE field in the KERNEL PARAMETERS (#8989.2) file. System administrators can also design a customized form with the same name as the local INPUT template that is displayed instead, terminal type setup permitting. In other words, to invoke a locally modified display, an INPUT template *must* exist. If a ScreenMan form by the same name also exists, an attempt is made to display the form before defaulting to the INPUT template.

REF: For more information on creating a local Edit User Characteristics form and template, see the *Kernel Installation Guide*.

For a sample form, see the "Edit User Characteristics Option" section.

3.4 Deactivating and Reactivating Users

Kernel provides options to deactivate and reactivate users on the **User Management** [XUSER] menu. When users no longer need access privileges, system administrators can partially or entirely close access to their account.

Figure 32: User Management Menu Options

```
SYSTEMS MANAGER MENU ...                                              [EVE]
User Management ...                                                 [XUSER]
   Deactivate a User                                            [XUSERDEACT]
   Purge Inactive Users' Attributes                           [XUSERPURGEATT]
   Reactivate a User                                            [XUSERREACT]
```

3.4.1 Deactivating Users

The **Deactivate a User** [XUSERDEACT] option lets you temporarily or permanently disable access for users. You can schedule termination of a user for a future date. The **Deactivate a User** [XUSERDEACT] option loads a ScreenMan form with the fields described in Table 5:

Table 5: Deactivate a User Option—Editable Fields/Attributes

Field/Attribute	Description
DISABLE USER	Setting the DISABLE USER field to **YES** prevents a user from signing on, but leaves all of their menus, keys, and other attributes (essentially the user's entire account) still enabled. It sets the DISUSER (#7) field in the user's NEW PERSON (#200) file to **YES**. You might want to use this feature to prevent access to your system by an external support person, except during pre-approved times (where you may want to monitor their actions). Setting DISUSER to **YES** prevents them from logging on to the system until you clear the field. If you set this field to **YES**, *do not set any other fields* in the Deactivate a User form (they only apply to terminating a user). Then, to re-enable access, use the **Reactivate a User** [XUSERREACT] option. **REF:** For a description of the **Reactivate a User** [XUSERREACT] option, see the "Reactivating Users" section.
TERMINATION DATE (#9.2)	Terminating a user is the way to formally deactivate a user (as opposed to temporarily disabling their account). Setting this date effectively terminates that user's account, effective from that date forward. The **Deactivate a User** [XUSERDEACT] option automatically performs the following steps when you deactivate a user: • Revokes the user's status as an authorized sender of any mail groups. • Revokes the user's status as a surrogate. • Revokes the user's status as a Secure Menu Delegation delegate. • Deletes the user's Access code, Verify code, Electronic Signature code, VA FileMan Access code (i.e., FILE MANAGER ACCESS CODE [#3] field), and Programmer Access code. • Deletes the user's menu templates. • Deletes the user's delegated options. • Purges the **^DISV** global on that CPU for that user. You can also decide whether all mail messages and all security keys for the account are deleted on the TERMINATION DATE with the final two fields in the **Deactivate a User** [XUSERDEACT] option (DELETE

Field/Attribute	Description
	ALL MAIL ACCESS and DELETE KEYS AT TERMINATION). If the user is expected to return to the facility and needs to have the user account reopened, security keys and mail could be retained. ℹ **REF:** For more information on cleaning up user access and privileges at termination, see the "XU USER TERMINATE Option" section in the "Signon/Security: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.
DELETE ALL MAIL ACCESS (#9.21)	Setting the DELETE ALL MAIL ACCESS field causes all mail messages for the user to be deleted when their account is terminated on the TERMINATION DATE.
DELETE KEYS AT TERMINATION (#9.22)	Setting the DELETE KEYS AT TERMINATION field causes all security keys for the user to be deleted at termination (except security keys marked "KEEP AT TERMINATE"). As discussed in the "Security Keys" section, the application developer can export a security key with the KEEP AT TERMINATE field set to **YES** in such a situation. The Provider security key, included with Kernel, has the flag set to **YES** for this purpose. Although a user may have been deactivated, it could be important to continue a processing activity that the user had authorized, based on privileges associated with a security key. A medical order could continue to hold an approved status, for example, even though the authorizing provider had been deactivated.

3.4.2 Automatically Deactivating Users

The **Automatic Deactivation of Users** [XUAUTODEACTIVATE] option finds all users in the NEW PERSON (#200) file with a TERMINATION DATE (#9.2) in the past, but who still have an Access code. In addition, it also looks to see if there are any users who have *not* signed on in the last "**n**" days.

ℹ **NOTE:** Kernel records all signons to VistA using appropriate user credentials via either of the following methods:

- Access and Verify codes.
- 2-Factor Authentication (2FA)—Digital certificate in a VA-approved smart card, such as the Personal Identification Verification (PIV) smart card plus a Personal Identification Number (PIN).

The **Automatic Deactivation of Users** [XUAUTODEACTIVATE] option terminates any users who fit these criteria. Any such users are users who had been scheduled for termination but were *not* terminated (usually because the task that should have terminated them did *not* run). It acts as a safety net to ensure that all users who were scheduled for termination are, in fact, terminated. It should be scheduled to run on a regular basis.

The **Automatic Deactivation of Users** [XUAUTODEACTIVATE] option is normally tasked to run daily via TaskMan.

REF: For *recommended* frequency of scheduling, see the *Kernel Installation Guide*.

Because the **Automatic Deactivation of Users** [XUAUTODEACTIVATE] option is *not* intended for interactive use, it is placed on the **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu.

3.4.2.1 Bulletin Notification

As of Kernel Patch XU*8.0*693, when the **Automatic Deactivation of Users** [XUAUTODEACTIVATE] option is used, it generates the **XUSERDEAC** MailMan bulletin, which is sent to the designated ISO SECURITY mail group in the bulletin's parameters. The Information Security Officers (ISOs) and their alternates are instructed to request membership in the ISO SECURITY mail group at their facility.

Figure 33: Sample XUSERDEAC MailMan Bulletin

```
Subj: XUSER DEACTIVATION  [#55995] 03/15/18@09:48  8 lines
From: XULASTNAME,ONEFIRSTNAME  In 'IN' basket.   Page 1
---------------------------------------------------------------------------

      User name : XULAST,ONEFIRST C
      Title     : DEVELOPER
      Service   : VHIT Field Office
      IEN       : 1039
      Station # : 662 SAN FRANCISCO

was deactivated on Mar 15, 2018.

Enter message action (in IN basket): Ignore//
```

The **XUSERDIS** MailMan bulletin was created to differentiate between DISUSERed and Deactivated users. Setting the DISUSER (#7) field in the NEW PERSON (#200) file to "**Yes**" for a user triggers the **XUSERDIS** bulletin. Setting the DISUSER (#7) field in the NEW PERSON (#200) file field to "**Yes**" only disables the user's ability to log on to the VistA system. It leaves all menus, security keys, and other attributes intact. The **XUSERDIS** bulletin is sent to the designated ISO SECURITY mail group in the bulletin's parameters. The data fields displayed in the **XUSERDIS** bulletin are the same as the **XUSERDEAC** bulletin.

Figure 34: Sample XUSERDIS MailMan Bulletin

```
Subj: USER DISUSERED  [#306499] 02/26/18@16:46  1 line
From: LASTNAME,FIRSTNAME (VHA OCC)  In 'IN' basket.   Page 1
---------------------------------------------------------------------------


      User name : LAST,TEST C
      Title     : DEVELOPER
      Service   : VHIT Field Office
      IEN       : 1039
      Station # : 662 SAN FRANCISCO

  was DISUSERED on Apr 05, 2018.

Enter message action (in IN basket): Ignore//
```

REF: For more information on the **XUSERDEAC** and **XUSERDIS** MailMan bulletins, see the “Bulletins” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual*.

3.4.2.2 Termination Process

The termination process does the following:

- Sets the DISUSER (#7) field in the NEW PERSON (#200) file to **YES** (1).
- Deletes the user’s Access code.
- Deletes the user’s security keys.
- Calls the XU USER TERMINATE protocol in the OPTION (#19) file so other applications can take any action they need.
- If the DELETE ALL MAIL ACCESS (#9.21) field in the NEW PERSON (#200) file is set to **YES**, then the user is also removed from the VistA MailMan system, which deletes their MailMan mail boxes and deletes them from any mail groups.

CAUTION: Kernel patch XU*8*645 created the XU645 parameter. It determines if a terminated user information should be purged from the system (Inspector General investigation request). When XU645 is set to YES, then data is deleted and the background AUTODEACTIVATE job purges those users who were previously terminated. The default value for XU645 is blank, which is equivalent to NO; it only deletes the user’s Access code and does *not* delete any other information or trigger any background jobs.

3.4.2.3 Academic Affiliation Waiver

The *VA Handbook 6500* page 60 (POLICY AND PROCEDURES, Technical Controls, Logical Access Controls), Item “d” states that accounts are automatically disabled if inactive for 30 days. This requirement is repeated in *VA Handbook 6500* Appendix D.

The Office of Academic Affiliation requested a waiver for the **30**-day disabling of inactive accounts asking it to be **90** days and the waiver was approved.

REF: A copy of the approved waiver is available as an attachment to Remedy Ticket #283028.

Kernel patch XU*8.0*514 added the ACADEMIC AFFILIATION WAIVER (#13) field to the KERNEL SYSTEM PARAMETERS (#8989.3) file. This field is used to control the LAST SIGN-ON DATE/TIME (#202) field in the NEW PERSON (#200) file. If the Office of Academic Affiliation waiver is applicable to a site, the site can set the ACADEMIC AFFILIATION WAIVER (#13) field to **YES** (1). The default for this field is **NULL**.

When the ACADEMIC AFFILIATION WAIVER (#13) field is set to **YES**, the users is only automatically disabled if they have been inactive for over **90** days (i.e., LAST SIGN-ON DATE/TIME is over **90** days). If it is *not* set, this option works as usual (i.e., **30**-day limit).

3.4.3 Purging Mail and Security Keys for Inactive Users

You can use the **Purge Inactive Users’ Attributes** [XUSERPURGEATT] option to clean up files. It removes all mailboxes, messages, mail groups, and security keys for users who have been terminated. If any of these users still retain Access codes, they are deleted.

This is particularly significant with mail. A mail message *cannot* be completely removed from a system until all recipients have deleted it from their mail baskets. If a user is no longer active, then it becomes unlikely that the message ever gets purged.

There are two modes of running this option. You can VERIFY the process for each user that the computer selects as eligible. If you choose *not* to verify the process for each user, then for every user with a *non*-future TERMINATION DATE, their set of security keys, mail groups, messages, and mail baskets are deleted.

3.4.4 Reactivating Users

You can use the **Reactivate a User** [XUSERREACT] option to re-enable access for a user who has either been terminated, or whose access has been temporarily disabled. To re-enable access for someone whose account is merely disabled (with the DISUSER field set to **YES**), use this option to simply clear the DISUSER field. Otherwise, using this option, you can fill in all the fields needed for an active account (i.e., FILE MANAGER ACCESS CODE [#3] field, PRIMARY MENU OPTION [#201], etc.).

When you reactivate a user, you are asked whether to deny access to old mail messages. If the reactivated user account is a less privileged account than previously, it may be appropriate to deny the user access to messages that were received in the user’s prior capacity. Even if that user’s mailbox was deleted at termination, once the user is reactivated, an old message would be delivered if responded to by another recipient.

3.5 User Management Menu

Kernel provides the **User Management Menu** [XUOPTUSER] located under the **Operations Management** [XUSITEMGR] menu. This menu provides a set of options for system administrators to monitor and support users logged onto the system. It includes the options shown in Figure 35:

Figure 35: User Management Menu Options

```
SYSTEMS MANAGER MENU ...                                                  [EVE]
Operations Management ...                                           [XUSITEMGR]
   User Management Menu ...                                         [XUOPTUSER]
   FIND   Find a user                                             [XU FINDUSER]
   PXY    Proxy User List                                   [XUSAP PROXY LIST]
          List users                                                [XUSERLIST]
          Print Sign-on Log                                         [XUSC LIST]
          Proxy (Connector) Detail Report        [XUSAP PROXY CONN DETAIL ALL]
          Proxy (Connector) Inquire              [XUSAP PROXY CONN DETAIL INQ]
          Release user                                               [XUSERREL]
          Remote Access User Sign-on Log                [XUSEC REMOTE ACCESS]
          User Inquiry                                               [XUSERINQ]
          User Status Report                                     [XUUSERSTATUS]
          Users with Foreign Visits                           [XUS VISIT USERS]
```

3.5.1 Find a User Option

The **Find a User** [XU FINDUSER] option is used to find a user who is currently signed on to the system in this UCI group. If you are on the same CPU as the user, this option also shows the menu path of the user. The option finds users based on the "CUR" cross-reference of the SIGN-ON LOG (#3.081) file.

3.5.2 Proxy User List Option

The **Proxy User List** [XUSAP PROXY LIST] option runs a report listing any users in the NEW PERSON (#200) file that have a USER CLASS (#9.5) field of APPLICATION PROXY or CONNECTOR PROXY.

3.5.3 List Users Option

The **List Users** [XUSERLIST] option lists all users known to the system.

3.5.4 Print Sign-on Log Option

The **Print Sign-on Log** [XUSC LIST] option prints out a Kernel sign-on log report (see Figure 36) that lists data values from fields in the SIGN-ON LOG (#3.081) file.

Table 6 lists the data displayed on the Kernel sign-on log report (see Figure 36):

Table 6: Kernel Sign-On Log Report Data Values

Report Field	File #3.081 Field Reference	Description
Sign-on time	DATE/TIME (#.001)	This is the date and time that the user signed onto the system. ℹ **NOTE:** To allow more than one signon per second the time can have values that show hundredth of a second.
ELAPSED TIME (MINUTES)	ELAPSED TIME (MINUTES) (#99)	This is the amount of time in minutes that the user has been signed onto the system.
USER	USER (#.01) Points to the NEW PERSON (#200) file.	This is the user name signed onto the system (i.e., LAST NAME,FIRST NAME).
$I	DEVICE $I (#1)	This is the **$I** device to which the user signed onto the system. This field holds the Hardware port name that the operating system (OS) can identify when referencing a port on a CPU. On layered systems where opening of host files is supported, this field can hold the host file name.
NODE NAME	NODE NAME (#10)	This is the VAX/VMS cluster node name or system name to which the user signed onto the system.
IPV6 ADDRESS	IPV6 ADDRESS (#100)	This is the IPV6 address from the calling system. Under the Dynamic Host Control Protocol (DHCP) Internet Protocol (IP) addresses are dynamically allocated, so more than one client could have used the same IP address over some time period. Also, under IPv6, each client could have more than one IP address. ℹ **NOTE:** IPv4 addresses are stored as IPv4-mapped IPv6 addresses, and all addresses are stored in expanded IPv6 format.

Report Field	File #3.081 Field Reference	Description
LOA	LEVEL OF ASSURANCE (#101)	This is the Level of Assurance (LOA) of the user's authentication into VistA. There are currently four levels defined by the National Institution of Standards and Technology Special Publication (NIST SP) 800-63-2 Electronic Authentication Guideline: • **Level 1—**No identity proofing requirement. This generally refers to a "self-asserted" user identity and is the lowest form of authentication. This form of authentication does *not* satisfy VA Handbook 6500 security requirements. • **Level 2—**Single factor authentication. This form of authentication includes username/password or, in the case of VistA, Access/Verify code authentication. • **Level 3—**Multi-factor authentication. This form of authentication includes VA 2-Factor Authentication (2FA) using smart cards (PKI certificates) and Personal Identification Number (PIN). • **Level 4—**Highest practical authentication assurance. At this level, in-person identity proofing (e.g., fingerprint or retinal scan) is used to authenticate and identify the user.
REMOTE APP	REMOTE APP (#18) Points to the REMOTE APPLICATION (#8994.5) file	The REMOTE APP (#18) field was added to the Kernel sign-on log report as of Kernel Patch XU*8.0*630. The data identifies how users are accessing VistA. For example, through any of the following applications: • JLV Application using National Health Information Network (NHIN). • VistA Applications (e.g., CPRS GUI, VistA Imaging VIX, etc.). • Terminal Emulator Software (e.g., Micro Focus® Reflection, Attachmate® Reflection, other terminal emulator, or generic default for a telnet/SSH interface). • Web Services.
CREDENTIAL TYPE	CREDENTIAL TYPE (#102)	This field contains the value of the type of credential used during VistA Kernel signon. In the past, it was assumed credentials were Access/Verify codes; now it lists other credentials being used: • AVCODES

Report Field	File #3.081 Field Reference	Description
		• SSOI **NOTE:** This field was added with Kernel Patch XU*8.0*701.
CREDENTIAL WARNINGS	CREDENTIAL WARNINGS (#103)	This field contains a list of credential verification failures that are considered warnings, when setting the STRICT TOKEN VALIDATION (#220) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file to **NO**, *non*-strict validation: • **CAFILE—**CA certificate chain file "**cache.cer**" has not been installed. • **DIGEST—**Token has been modified. • **SIGNATURE—**Token might have been reformatted and signature verification failed. • **EXPIRED—**Token has expired. **NOTE:** This field was added with Kernel Patch XU*8.0*701.

Figure 36: Sample Kernel Sign-On Log Report

```
SIGN-ON LOG List                                     NOV 06, 2019@15:03   PAGE 3
                         ELAPSED
                            TIME
Sign-on time           (MINUTES)  USER               $I         NODE NAME
  IPV6 ADDRESS                                LOA  REMOTE APP
  CREDENTIAL
  TYPE
  CREDENTIAL WARNINGS
--------------------------------------------------------------------------------
OCT 30,2019@05:56:19          3   XUUSER,NINE    /dev/pts/  vhaausdhct033
  0000:0000:0000:0000:0000:FFFF:0AEC:84EF   2    TERMINAL EMULATOR
  AVCODES
OCT 29,2019@12:31:26          6   XUUSER,SEVEN      /dev/pts/  vhaausdhct033
  0000:0000:0000:0000:0000:FFFF:0AEC:84E4   3    MICRO FOCUS REFLECTION
  SSOI
OCT 29,2019@11:22:04         10   XUUSER,TEN         /dev/pts/  vhaausdhct033
  0000:0000:0000:0000:0000:FFFF:0A06:0226   3    REMOTE APP1
  SSOI
  CAFILE;SIGNATURE;
OCT 29,2019@12:11:36         26   XUUSER,ELEVEN      /dev/pts/  vhaausdhct033
  0000:0000:0000:0000:0000:FFFF:0A06:0226   3    REMOTE APP2
  SSOI
  SIGNATURE;
```

3.5.5 Proxy (Connector) Detail Report Option

The **Proxy (Connector) Detail Report** [XUSAP PROXY CONN DETAIL ALL] option provides information about CONNECTOR PROXY accounts for the purposes of:

- Monitoring compliance with the 3-year mandate (per VA Handbook 6500) to expire/change Verify codes for service accounts.
- Reporting any misconfigured CONNECTOR PROXY accounts.
- Listing account activity to help determine whether accounts are active and are being accessed from which remote locations.

When running the report, the following options determine how much additional content is listed for each account:

- Check/display connector proxy fields? YES/NO (checks for misconfigured accounts).
- Scan sign-on log for connector proxy activity? YES/NO (lists account activity).

Possible categorizations for whether accounts are reported as "Compliant w/3-year Service Account Mandate?" are:

- YES (account is compliant).
- *** NO <---- MUST FIX *** (date created and date verify code last changed > **3 years** in the past).
- No, but user *not* active.
- UNABLE TO DETERMINE (until patch XU*8.0*574, date verify code last changed for Connector Proxy accounts was incorrectly recorded as 4/10/2005)
- Unable to determine but *not* active.

If an account's Date Verify Code Last Changed is listed as "(changed but date *not* recorded)", that means the "fake" **4/10/2005**" date is present, and unless the account was created within the last **3** years, it is impossible to determine if the account complies with the **3**-year mandate.

Also, if there is a value in the XUS LOGON ATTEMPT COUNT field, that value is displayed, as it could indicate a remote system attempting to connect and failing with an invalid Verify code.

If the option to “Check/display connector proxy fields?” is selected, the following checks are performed:

- Warnings: (any field listed in the warning section should *not* be populated. However, before changing, consult the National Help Desk or Customer Support as some applications may (currently) be depending (incorrectly) on a misconfigured connector configuration.)
- Values for other fields allowed/expected: (field normally populated for connector proxies).
- Other Fields Populated (not expected fields, but *not* problematic either).
- Other Multiples Populated (not expected, but *not* problematic either).

If the option to “Scan sign-on log for connector proxy activity?” is selected, the report scans the sign-on log for all signon activity associated with the account. Any activity found is displayed, organized by client IP address, and within IP address, by date of signon.

The purpose of this report section is to help sites determine the following:

- Which accounts are active.
- Which external systems (by IP address) are logging onto the site with the specified account.

This helps determine which remote applications a change to the account (e.g., Verify code change) might impact, and it also helps a site determine whether too many remote applications/data centers are using the same account (which could result in a more widespread service disruption if an account *must* be changed).

NOTE: This option can be scheduled.

3.5.6 Proxy (Connector) Inquire Option

The **Proxy (Connector) Inquire** [XUSAP PROXY CONN DETAIL INQ] option provides information about CONNECTOR PROXY accounts for the same purposes as the Proxy (Connector) Detail Report Option; however, it allows the selection of a specific NEW PERSON (#200) file CONNECTOR PROXY entry.

3.5.7 Release user Option

If multiple signons are prohibited, problems can occur if users experience an abnormal exit such that the signon record *cannot* be cleared. System administrators can use the **Release user** [XUSERREL] option to remedy the problem for individual users.

To clear all users on startup, schedule the **Clear all users at startup** [XUSER-CLEAR-ALL] option.

3.5.8 Remote Access User Sign-on Log Option

The **Remote Access User Sign-on Log** [XUSEC REMOTE ACCESS] option prints sign-on log entries from remote users (VISITORS) that have been authenticated on an external system (usually another VistA server) using Broker Security Enhancement (BSE) or the (deprecated) Medical Domain Web Service (MDWS) visitor access.

The report shows:

- Remote Site Name.
- Date of First Visit.
- Date of Last Visit.

BSE allows users to be validated through 2-Factor Authentication (2FA) or the traditional VistA Access and Verify codes on their home system and then carry that authentication to other VistA systems. A packet of information is retrieved from the authenticating (home) site and is entered in the NEW PERSON (#200) file, so that a trace to the original authentication can be made.

3.5.9 User Inquiry Option

The **User Inquiry Option** [XUSERINQ] option displays various attributes of a specified user. If the user is currently signed on, it displays the following:

- Job and Device numbers.
- Signon time.
- What option is being executed.

Otherwise, it displays the last signon time. It also displays which security keys are held by the user.

3.5.10 User Status Report Option

The **User Status Report** [XUUSERSTATUS] option produces a report of the users currently signed on to this CPU and UCI. It shows the option each user is running and when they signed on, as well as their device and job numbers.

3.5.11 Users with Foreign Visits Option

The **Users with Foreign Visits** [XUS VISIT USERS] option shows NEW PERSON (#200) file entries that have been VISITORS to this site using Broker Security Enhancement (BSE) or the (deprecated) Medical Domain Web Service (MDWS) visitor access.

3.6 Signon Audits

Signon events are recorded in the SIGN-ON LOG (#3.081) file. Statistics, such as the time of access and the user's identity, are stored for audit purposes. If the user exits normally (is *not* "bumped" off the system), the signon record includes the time of exit. If the user exits abnormally with an error or enters programmer mode, the signon record *cannot* include a time of exit.

Information about signon activity can be reviewed with options on the **Systems Manager Menu** [EVE] and **Operations Management** [XUSITEMGR] menus.

The SIGN-ON LOG (#3.081) file is purged with the **Purge Sign-On log** [XUSCZONK] option that should be tasked to run on a regular schedule (e.g., every night). This option *cannot* be reached from Menu Manager; like other options that should only be queued, it is on the **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu.

As of Kernel Patch XU*8.0*756, the following functionality enhancements were made:

- Added the **SIGN-ON LOG RETENTION (#221)** parameter field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. This parameter determines the number of entries (number of days) to retain data in the Kernel sign-on log [SIGN-ON LOG (#3.081) file]. Larger values will consume more disk space; so, sites should evaluate the impact on data storage *before* changing the default value of **365 days**.
- Added the "**SIGN-ON LOG RETENTION**" field to the **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option form (Figure 12). This new form field allows the user to set the value of the **SIGN-ON LOG RETENTION (#221)** parameter field in the KERNEL SYSTEM PARAMETERS (#8989.3) file.
- Enhanced the **SCPURG^XUSPURGE** routine. It updated the **Purge Sign-on Log** [XUSCZONK] option so that it protects sign-on log entries for at least the number of days specified by the ISO or the default **365 days** as defined in the **SIGN-ON LOG RETENTION (#221)** parameter field in the KERNEL SYSTEM PARAMETERS (#8989.3) file.

3.6.1 Signon Statistics

Statistics about active sessions can be obtained with the **CPU/Service/User/Device Stats** [XUSTAT] option. This option permits sorting by CPU, by the user's Service/Section (e.g., MAS) by individual users, or by particular devices.

Figure 37: CPU/Service/User/Device Stats Option

```
SYSTEMS MANAGER MENU ...                                            [EVE]
Operations Management ...                                     [XUSITEMGR]
   CPU/Service/User/Device Stats                                 [XUSTAT]
```

3.6.2 Failed Access Attempts Audit

3.6.2.1 Access/Verify Codes Authentication

When a user enters invalid Access and Verify code pairs, the number of attempts is recorded, and the device appears to lock after the site parameter limit of failed access attempts is reached as designated in the DEFAULT # OF ATTEMPTS field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. After this point, Signon/Security continues to record what the user types (but only to create a record in the FAILED ACCESS ATTEMPTS LOG [#3.05] file).

REF: For more information on the DEFAULT # OF ATTEMPTS field in the KERNEL SYSTEM PARAMETERS (#8989.3) file, see Section 3.1.2.1, "Signon Attempts and Device Lock-out Times."

If a valid Access code is entered, Signon/Security can link the attempt with a known user and records that user's name in the log. Since it is a valid code, its text is *not* recorded in the log. The text of subsequently entered invalid Verify codes can, however, be recorded as clues to the source of the access attempt. If the Access code is *not* valid, a user's name *cannot* be associated but the text of the attempt can be recorded.

The log also records the following:

- Time of day.
- Device used.
- CPU/UCI location.

3.6.2.2 PIV 2-Factor Authentication (2FA)

With Personal Identity Verification (PIV) 2-Factor Authentication (2FA) the following occurs:

1. User gets ***one*** attempt with PIV card authentication. If it fails, the failure is recorded in the FAILED ACCESS ATTEMPTS LOG [#3.05] file and the authentication process automatically reverts to using the Access and Verify codes.

   NOTE: Prior to Kernel Patch XU*8*701, these failures were *not* being recorded and it was very difficult for sites to troubleshoot PIV card failures.

2. User gets ***as many attempts*** using Access and Verify codes as they are configured in the DEFAULT # OF ATTEMPTS field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. When the number is reached, the failure is recorded in the FAILED ACCESS ATTEMPTS LOG [#3.05] file.

NOTE: When troubleshooting issues related to PIV card authentication or other related credentials like Access/Verify codes, the site can "temporarily" set the value of the

FAILED ACCESS ATTEMPTS field in the **Establish System Audit Parameters** [XUAUDIT] option to the following value:

ALL DEVICES/TEXT RECORDED

CAUTION: Due to the unexpected number of PIV authentication failures, your site may need to increase the frequency of running the Failed Access Attempts Log Purge [XUFPURGE] option for the FAILED ACCESS ATTEMPTS LOG [#3.05] file.

3.6.2.2.1 PIV 2-Factor Authentication Failures

Failures recorded when using the PIV 2-Factor Authentication (2FA) authentication contain the value of the **SecID** and name of the user identified in the SSOi token derived from PIV 2FA. Other failures in validating the SSOi token or other tokens are also listed, as shown in Figure 38.

Figure 38: Failed Access Attempts Log Report

```
LOG OF USER FAILED ACCESS LIST  NOV 6,2019         3:38 PM    PAGE  1
--------------------------------------------------------------------------------
              *** USER NAME:

DATE/TIME OF ATTEMPT:  NOV 6,2019@15:37:40
    NUMBER OF ATTEMPTS:     3          TYPE OF FAILED ATTEMPT:  ACCESS
    CPU:  KRN      UCI:  KRN           DEVICE:  TELNET (LINUX)  (10.6.2.38)

    TEXT ENTERED:
    Access: XXXXX
    Access: YYYYYY
    Access: ZZZZZZ

              *** USER NAME:

DATE/TIME OF ATTEMPT:  NOV 6,2019@15:37:13
    NUMBER OF ATTEMPTS:     1          TYPE OF FAILED ATTEMPT:  SSOI
    CPU:  KRN      UCI:  KRN           DEVICE:  TELNET (LINUX)  (10.6.2.38)

    TEXT ENTERED:
    NON-STRICT Failed-verifications: CAFILE;SECID;SIGNATURE;   ERROR:User not
found. SECID not linked to existing user, SECID:1005555055 NAME:TEN XUUSER
```

3.6.2.2.2 Failed Verifications for Token (SSOi)

The following is a list of some failed verification that you may encounter:

- **SECID**—VistA Kernel could not match the SecID in the token to an entry in the NEW PERSON (# 200) file.
- **EXPIRED—**Token has expired.
- **CAFILE—**CA certificate chain file "**cache.cer**" has not been installed.

- **DIGEST**—Token has been modified.
- **SIGNATURE**—Token might have been reformatted and signature verification failed.
- **EXPIRED**—Token has expired.

3.6.2.3 Kernel Signon Auditing Files

Table 7: Kernel Signon Auditing Files

File	Global Location	Set Parameters	Display Parameters	Initiate/ Terminate	Print Reports	Purge Logs
SIGN-ON LOG (#3.081)	**^XUSEC(0,**	Predefined	N/A	Always done	Print Sign-on Log [XUSC LIST]	Purge Sign-on Log [XUSCZONK]
FAILED ACCESS ATTEMPTS LOG (#3.05)	**^%ZUA(3.05,**	Establish System Audit Parameters [XUAUDIT]	Display the Kernel Audit Parameters [XU-SPY-SHOW]	On/Off switch	Devices: Device Failed Access Attempts [XUFDEV] Users: User Failed Access Attempts [XUFDISP]	Failed Access Attempts Log Purge [XUFPURGE]
OLD ACCESS AND VERIFY CODES (#200 XREF)	**^VA(200,**	Predefined	N/A	Always done	N/A	Purge Log of Old Access and Verify Codes [XUSERAOLD]

3.6.3 Purge Old Access and Verify Codes

Figure 39: Purge Log of Old Access and Verify Codes Option

```
SYSTEMS MANAGER MENU ...                                                     [EVE]
User Management ...                                                        [XUSER]
   Purge Log of Old Access and Verify Codes                            [XUSERAOLD]
```

The **Purge Log of Old Access and Verify Codes** [XUSERAOLD] option purges all inactive Access and Verify codes, which allows for the recycling of codes. Old Access and Verify codes are stored so that users *cannot* pick a previously used code when required to choose a new code. If old codes are stored indefinitely, though, it may become difficult for users to invent new codes. When you use this option interactively, you can purge codes older than a retention period you specify, from **7** to **90** days. When scheduled, the retention period defaults to **90** days, but can be changed to anything from **30** to **90** days by putting the number of days in the TASK PARAMETERS field of the OPTION SCHEDULING (#19.2) file.

The log of Access codes is stored in the whole-file **AOLD** cross-reference of the NEW PERSON (#200) file. The log of Verify codes is stored per user in the **VOLD** cross-reference of the NEW PERSON (#200) file, *not* a whole-file cross-reference). Thus, Verify codes are *not* necessarily unique between users, while Access codes are.

4 File Access Security

The File Access Security system is an optional Kernel module. It provides an enhanced security mechanism for controlling user access to VA FileMan files.

REF: For more information on File Access Security, see the *VA FileMan (Version 22.0) and Kernel (Version 8.0) File Access Security* supplemental documentation located on the VA Software Document Library (VDL) at: VDL VA FileMan Application Documents

4.1 User Interface

As a user, you typically access VistA data by use of application options. You enter data into files and retrieve information from files through the menu options within the software. Except under a few unusual circumstances, your use of the system is *not* affected by the File Access Security system. If you need to work directly with files by using VA FileMan options, however, you are affected.

VA FileMan options provide direct access to data files. Figure 40 lists some sample VA FileMan options:

Figure 40: Sample VA FileMan Menu Options

```
Select VA FileMan Option: ?

          Enter or Edit File Entries                                  [DIEDIT]
          Print File Entries                                         [DIPRINT]
          Search File Entries                                       [DISEARCH]
          Inquire to File Entries                                  [DIINQUIRE]
```

If the File Access Security system is implemented, the only files you can access directly through VA FileMan options are those listed in your ACCESSIBLE FILE (#32) Multiple field in the NEW PERSON (#200) file. System administrators grant file access by using a submenu on the **User Management** [XUSER] menu.

There are six levels of File Access Security properties (listed alphabetically):

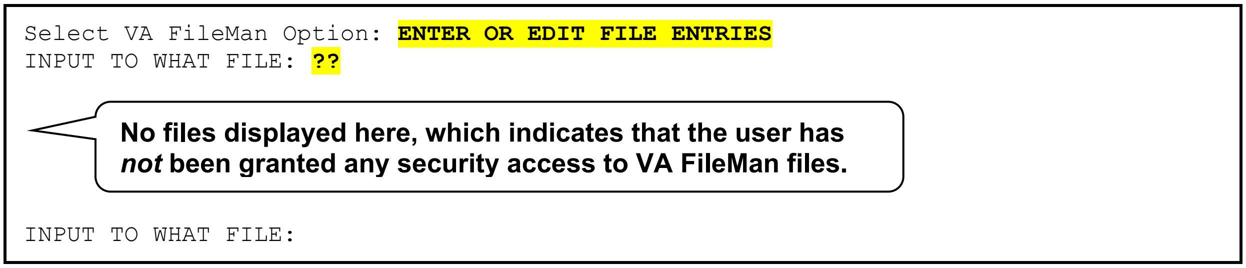
- **AUDIT**
- **DATA DICTIONARY (“DD”)**
- **DELETE (“DEL”)**
- **LAYGO**
- **READ (“RD”)**
- **WRITE (“WR”)**

REF: These File Access Security level properties are described in Table 8.

Each level of access is granted as **YES** or **NO**. If the File Access Security system is implemented, file access is controlled by these **YES/NO** flags, *not* by the matching of your FILE MANAGER ACCESS CODE (#3) field string in the NEW PERSON (#200) file with security placed on the file.

If you have *not* been granted any security access to VA FileMan files, entering two question marks (**??**) when prompted for a file name/number shows no files to access:

Figure 41: User has *Not* been Granted Security Access to any VA FileMan Files—Sample User Dialog



```
Select VA FileMan Option: ENTER OR EDIT FILE ENTRIES
INPUT TO WHAT FILE: ??
```

No files displayed here, which indicates that the user has *not* been granted any security access to VA FileMan files.

```
INPUT TO WHAT FILE:
```

In this case, you need to contact the system administrators to get access to the VA FileMan files you need.

File Access Security is also invoked when an option uses VA FileMan's Line Editor. In particular, the Transfer Lines from Another Document option, which is hard-coded in the **DIWE** routine and is also referred to as the Line Editor, does *not* permit access to other word-processing documents in the current file or other files unless **READ** access to that file has been explicitly granted. If you need to transfer text from other files using the Line Editor, contact the system administrators to request access to those files.

4.2 System Management

Prior to introduction of the File Access Security system, user access to VA FileMan files through VA FileMan options was controlled by matching a character in a user's FILE MANAGER ACCESS CODE (#3) field [the **DUZ(0)** string] in the NEW PERSON (#200) file with a character in the file's top level file security fields.

Kernel's optional File Access Security system uses a different method. It allows you to control access to files for any user using VA FileMan options directly. Access is granted (or denied) by adding (or removing) a file from a user's ACCESSIBLE FILE (#32) Multiple field in their NEW PERSON (#200) file entry.

The File Access Security system does *not* affect access to files through *non*-VA FileMan options; security in this case is managed by controlling the availability of the option.

REF: For exceptions, see the "When is File Access Security Checked?" section.

If a user's **DUZ(0)** is set to the at-sign (@; Programmer access), VA FileMan options allow complete file access. If it is set to anything else (except the caret [^]), VA FileMan options use the ACCESSIBLE FILE (#32) Multiple field specifications in the NEW PERSON (#200) file to grant varying levels of file access.

NOTE: The caret (^) overrides the at-sign (@; Programmer access).

This higher degree of control over a user's file access comes at a price, because it requires more management on the system administrator's part to provide each user access to the files to which they need access. However, the payoff in using the File Access Security system is in enhanced control and security for VA FileMan files.

4.2.1 When is File Access Security Checked?

When using VA FileMan options, access to files through the File Access Security system is checked.

When initially accessing data in a file through software options (e.g., options using VA FileMan Application Program Interfaces [APIs]), File Access Security is *not* checked. File Access Security is checked, however, when calling the following VA FileMan APIs:

- **^DIC calls—**Adding an entry to the top level of a file (i.e., **LAYGO** access)
- **^DIE calls—**Deleting an entry at the top level of a file (i.e., **DELETE** access).

Developers can bypass these **LAYGO** and **DELETE** access checks using the following variables, respectively:

- **DLAYGO**
- **DIDEL**

When accessing data through software options, File Access Security is also checked when a file is navigated to from another file (i.e., **READ**, **WRITE**, **DELETE**, and **LAYGO** access). Currently, there is no way for developers to override access checks when navigating to a file from another file, so explicit access to files navigated to/from an application option *must* be granted by the system administrators.

4.2.2 What in VA FileMan is Still Protected by the File Manager Access Code?

When the File Access Security system is enabled, access to templates (e.g., INPUT, PRINT, SORT, etc.) is denied when using VA FileMan options; if the user's **DUZ(0)** string does *not* contain a matching character. Similarly, when editing fields via VA FileMan's **Enter or Edit File Entries** [DIEDIT] option, the **DUZ(0)** matching process is invoked to permit or deny editing for protected fields. The **DUZ(0)** value is also checked by some *non*-VA FileMan applications. Finally, if a user's **DUZ(0)** is @, they are allowed complete access to all files.

4.2.3 Purpose for Granting File Access

System administrators are responsible for granting file access. They *must* determine each user's needs and assign an appropriate degree of access authority. Too much access may risk the security of your system, while too little may inhibit productive activity.

What is the purpose of File Access Security? Why bother specifying who has access to which files? The answer is threefold:

- To monitor the use of VA FileMan.
- To regulate the extent of VA FileMan access from among six levels of security that allow **AUDIT**, **DATA DICTIONARY ("DD")**, **DELETE ("DEL")**, **LAYGO**, **READ ("RD")**, or **WRITE ("WR")** access.

  REF: These File Access Security level properties are described in Table 8.

- To reserve **DUZ(0)**, the FILE MANAGER ACCESS CODE (#3) field, as a security measure to protect just templates and fields, *not* files, from VA FileMan options.

With file access security, it is possible to know who has access to which files and what kind of access they have. This information can also be retrieved by user or by file. In addition, privileges can also be entirely restricted for an individual user or for a single file that may contain sensitive information.

4.2.4 Who Needs File Access?

You need to grant File Access Security in the following cases:

- A user needs to access files directly through VA FileMan options.
- Within an application option, VA FileMan is used to navigate from one file to another.
- Within an application option that calls the ^DIE API to edit a file entry; a user is unable to add or delete entries in a pointed-to file.
- Within an application option that calls the ^DIE or ^DIC APIs to edit a file entry; a user is unable to add or delete entries in the primary file (because the application did *not* set the **DLAYGO** or **DIDEL** variables).
- A user needs to use VA FileMan's Line Editor's Transfer Lines from Another Document option.

Application developers can document which files need to be granted to whom, or they can modify their code or data dictionary (DD) specifications to allow access.

4.2.5 Levels of File Access Security

There are six file access security properties involved with File Access Security. If a file access security property is *not* defined (i.e., the value is **NULL**), the VA FileMan exported menu options for that property are *not* open to full access for users.

REF: Table 8 is taken from the *VA FileMan (Version 22.0) and Kernel (Version 8.0) File Access Security* supplemental documentation located on the VA Software Document Library (VDL) at: VDL VA FileMan Application Documents

Table 8: File Access—Security Level Properties

Access	Security Property Description	Property Location (Classic VA FileMan)
AUDIT	The **AUDIT** security property controls the setting of auditing characteristics and the deletion of audit trails. This property only deals with the auditing of data and *not* 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: • **Fields Being Audited** [DIAUDITED FIELDS] • **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]	^DIC(<file number>,0,”AUDIT”)=<value>
DATA DICTIONARY (“DD”)	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: • **Modify File Attributes** [DIMODIFY] • **Utility Functions** [DIUTILITY]	^DIC(<file number>,0,”DD”)=<value>

Access	Security Property Description	Property Location (Classic VA FileMan)
	- **Data Dictionary Utilities** [DI DDU] For example, to use the **Map Pointer Relations** [DI DDMAP] option, **DD** access is needed to the PACKAGE (#9.4) file and to the files one selects for mapping.	
DELETE (“DEL”)	The **DELETE** security property controls who can delete an existing record that is contained within the file. It does *not* permit deletion of the file or any of its attribute fields. Examples of the VA FileMan options that this property controls are as follows: - **Enter or Edit File Entries** [DIEDIT] - **Transfer Entries** [DITRANSFER]	^DIC(<file number>,0,”DEL”)=<value>
LAYGO	The **LAYGO** (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: - **Enter or Edit File Entries** [DIEDIT] ℹ **NOTE:** You *must* have **LAYGO** and **WRITE** access to a file to add new entries. In addition, you *must* have **WRITE** access at the field level for all required identifier fields.	^DIC(<file number>,0,”LAYGO”)=<value>
READ (“RD”)	The **READ** 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: - **Print File Entries** [DIPRINT] - **Search File Entries** [DISEARCH] - **Inquire to File Entries** [DIINQUIRE] - **Statistics** [DISTATISTICS] - **List File Attributes** [DILIST]	^DIC(<file number>,0,”RD”)=<value>

Access	Security Property Description	Property Location (Classic VA FileMan)
	• **Transfer Entries** [DITRANSFER] To transfer text, the user needs **READ** access to the file from which text is being transferred. Similarly, **WRITE** access is needed for the file to which entries are being transferred with this option. ℹ **NOTE: READ** access is also required to use some of the Filegram and Audit options.	
WRITE (“WR”)	The **WRITE** security property controls who can alter data in an existing record that is contained within the file. It does *not* permit the adding of new entries to the file. Examples of the VA FileMan options that this property controls are as follows: • **Enter or Edit File Entries** [DIEDIT] • **Transfer Entries** [DITRANSFER] To transfer text, the user needs **READ** access to the file from which text is being transferred. Similarly, **WRITE** access is needed for the file to which entries are being transferred with this option.	^DIC(<file number>,0,”WR”)=<value>

Any or all of these six levels of access can be enabled for each of the user’s accessible files. This is done by changing the field value from **NULL** to **YES**. This flag is overridden for developers whose **DUZ(0)=@**.

Granting the **READ**, **WRITE**, **DELETE**, and **LAYGO** levels of access permits adding and deleting file entries as well as editing their attribute field data values. This is true unless the attribute field has been protected. If so (i.e., if there is **READ**, **WRITE**, or **DELETE** protection within the data dictionary [DD] for a given field), the user’s FILE MANAGER ACCESS CODE (#3) field, **DUZ(0)**, is checked. Access is denied if the user’s **DUZ(0)** does *not* contain a character matching the field protection. Again, **DUZ(0)=@** overrides this restriction.

The **DATA DICTIONARY ("DD")** and **AUDIT** levels of access pertain to the structure of the file itself. While this provides a generous scope for VA FileMan data dictionary (DD) modification, it falls short of, for example, deleting a field protected with the at-sign (**@**; Programmer access).

The same applies to templates. If the template is protected, the user who has access to the file does *not* have access to the template from VA FileMan options unless there is a match in the **DUZ(0)** character string.

4.2.6 Audit Access to Files

Audit privileges might be granted to advanced VA FileMan users who are interested in developing new audit capabilities. With **AUDIT** access, which *must* be accompanied by **DD** access, VA FileMan's **Modify File Attributes** [DIMODIFY] option can be used to set an audit flag for a particular field within a file. This access does *not* include setting audit conditions with M code, which is reserved for users with a FILE MANAGER ACCESS CODE (#3) field containing **@**.

The data values for attribute fields can be recorded in the AUDIT (#1.1) file by setting an audit flag in the data dictionary (DD) for that field. For example, the SSN field in the PATIENT (#2) file could be audited. There are two choices for the audit in the AUDIT (#1.1) file:

- An entry can be made when a value is entered or changed.
- An entry can be made *only* when the value is changed (i.e., edited or deleted).

The second method may be all that's needed. In the SSN example, you would monitor just the circumstances of the change, *not* of the initial SSN assignment.

To display the results of the audit, your **DUZ(0)** *must* equal the at-sign (**@**; Programmer access). Then, you can query the AUDIT (#1.1) file in the usual way with VA FileMan's **Inquire to File Entries** [DIINQUIRE] option.

4.2.7 How to Grant File Access

System administrators specify the particular files and levels of access for users. The **File Access Security** [XUFILEACCESS] menu, on the **User Management** [XUSER] menu, provides options to grant file access security. These options edit the ACCESSIBLE FILE (#32) Multiple field in the NEW PERSON (#200) file.

The options for granting file access privileges fall into three functional categories:

- **EDITING—**To assign file access to an individual user or a group of users. One user's profile can also be duplicated or copied to another user or group of users. To simplify adding files, number ranges can be specified.
- **LISTING—**To display one user's profile, a name-sorted list of all user's profiles, or a file or range of files with associated users and the access levels of each.
- **RESTRICTING—**To entirely limit access by user or by file, or to delete a range of files for a user or group of users.

The options are designed to facilitate queries by user or by file. You can add or delete file access for one user or for many users. Or, you can begin with the file and list users with access or restrict access.

4.2.8 Using the File Access Options

The **File Access Security** [XUFILEACCESS] menu options are shown in Figure 42.

Figure 42: File Access Security Menu Options

```
SYSTEMS MANAGER MENU ...                                                      [EVE]
User Management ...                                                         [XUSER]
   File Access Security ...                                          [XUFILEACCESS]
      Grant Users’ Access to a Set of Files                           [XUFILEGRANT]
      Copy One User’s File Access to Others                            [XUFILECOPY]
      Single file add/delete for a user                           [XUFILESINGLEADD]
      Inquiry to a User’s File Access                               [XUFILEINQUIRY]
      List Access to Files by File number                              [XUFILELIST]
      Print Users Files                                               [XUFILEPRINT]
      Delete Users’ Access to a Set of Files                      [XUFILESETDELETE]
      Remove All Access from a Single User                        [XUFILEREMOVEALL]
      Take away All access to a File                                 [XUFILEDELETE]
      Assign/Delete a File Range                                [XUFILERANGEASSIGN]
```

When using options on the **File Access Security** [XUFILEACCESS] menu, you may have the following questions:

- What is the **DUZ#** that appears next to the user’s name?
- How is a range of file numbers specified?
- What are the queuing questions all about?

4.2.8.1 Understanding DUZ (User Number)

When listing the file accesses by user or by file, the user’s name is followed by a number in parentheses. The heading indicates that this is the “User #,” which is the same as the **DUZ#**.

Once the user enters an Access and Verify code, Kernel’s Signon/Security uses the **DUZ** variable to identify an entry in the NEW PERSON (#200) file. It *must* be a unique identifier, so the user’s name does *not* work. Instead, the Internal Entry Number (IEN) is used. That is what becomes the value of **DUZ**.

NOTE: Some users have low numbers while others have high ones. This simply indicates the order their names were entered into the NEW PERSON (#200) file. Users with low numbers are often people who began using the system some years ago, while users with high numbers tend to be recent entries in the file.

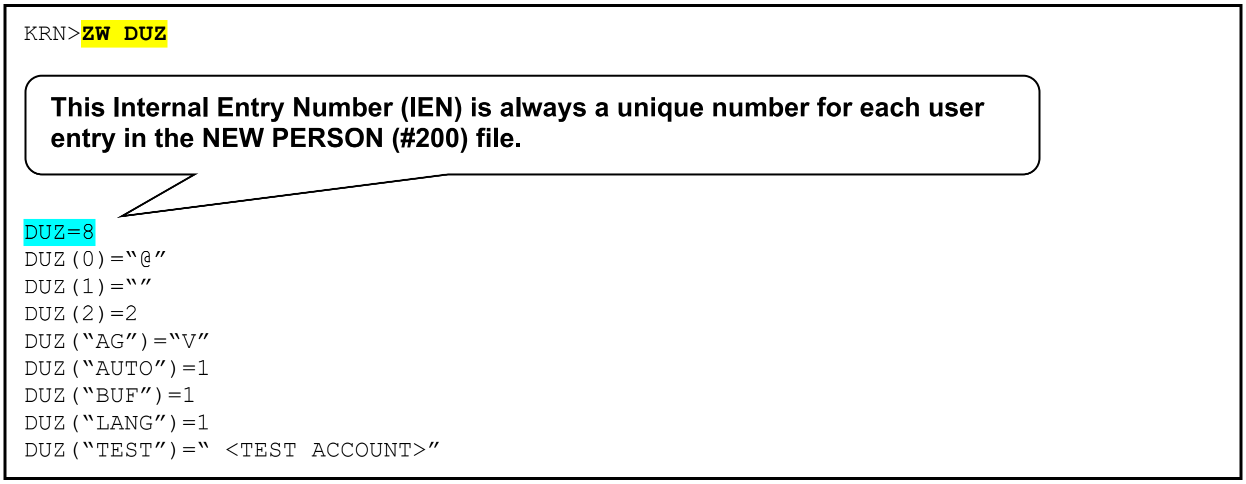
DUZ is a local variable array that identifies the user who has signed onto the system. It is the Internal Entry Number (IEN) for the user in the NEW PERSON (#200) file. Besides the unique IEN, the **DUZ** array contains other variables specific to the signed-on user, which are listed in Table 9:

Table 9: DUZ Array Variables

Variable	Description
DUZ(0)	This variable stores the level of Programmer access (i.e., VA FileMan Access Code) of the user at signon (e.g., **@**). This variable is derived from the value stored in the FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON (#200) file.
DUZ(1)	This variable is obsolete; it is always set to **NULL**.
DUZ(2)	If a user is associated with more than one institution (division), the user is prompted at signon to select a division. This variable is set to the appropriate value. This variable is derived from the values stored in the DIVISION (#16) Multiple field in the NEW PERSON (#200) file. This field points to the INSTITUTION (#4) file.
DUZ("AG")	This variable stores the agency code at signon (e.g., V = VA). This variable is derived from the value stored in the AGENCY CODE (#9) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. This value is a defined Set of Codes.
DUZ("AUTHENTICATION")	This variable stores the method used to authenticate the user. Examples include "**ASHTOKEN**", "**AVCODES**", "**BSETOKEN**", "**CCOWTOKEN**", "**SSOI**", "**SSOE**", "**NHIN**", "**NONE**", and "**XUP**".
DUZ("AUTO")	Menu Manager uses this variable to control whether all items on a menu are presented automatically after each cycle through the menu system. This variable stores the user's menu display preference at signon (e.g., **1** = Auto Generate Menus). This variable is derived from the value stored in the AUTO MENU (#.06) field in the NEW PERSON (#200) file.
DUZ("BUF")	This variable stores the user's type ahead (buffer) preference (e.g., 1 = Allowed). This variable is derived from the value stored in the TYPE-AHEAD (#.09) field in the NEW PERSON (#200) file.
DUZ("LANG")	This variable stores the display language as it is stored in the LANGUAGE (#.01) field in the LANGUAGE (#.85) file. VA FileMan uses this setting to enable the display of language-specific dates and times, numeric formats, and dialogs. VA FileMan currently distributes only the English language entry for this file (entry number 1).

Variable	Description
	The LANGUAGE (#.01) field in the LANGUAGE (#.85) file is pointed to by the following: • LANGUAGE (#.01) field of the TRANSLATION (#.847) subfield of the DIALOG (#.84) file. • LANGUAGE (#200.07) field in the NEW PERSON (#200) file. • DEFAULT LANGUAGE (#207) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file, which overrides the setting of the LANGUAGE (#200.07) field.
DUZ("LOA")	This variable records the "Level of Assurance" (LOA) of the user's authentication and identity. Four levels are currently defined by National Institution of Standards and Technology Special Publication (NIST SP) 800-63-2 Electronic Authentication Guideline: • **Level 1—**No identity proofing requirement. This generally refers to a "self-asserted" user identity and is the lowest form of authentication. This form of authentication does *not* satisfy VA HANDBOOK 6500 security requirements. Application developers may choose to programmatically deny access to sensitive data if a user's LOA equals "1". • **Level 2—**Single factor authentication. This form of authentication includes username/password or, in the case of VistA, Access/Verify code authentication. • **Level 3—**Multi-factor authentication. This form of authentication includes VA 2-Factor Authentication (2FA) using smart cards (PKI certificates) and Personal Identification Number (PIN). • **Level 4—**The highest practical authentication assurance. At this level, in-person identity proofing such as fingerprint or retinal scan is used to authenticate and identify the user.
DUZ("REMAPP")	This variable is used to identify an external client application whenever possible. Examples include "**BMS**", "**CAPRI**", "**MDWS**", "**NUMI**", "**VISTA IMAGING**", and others. The information is currently obtained from the REMOTE APPLICATION (#8994.5) file, but plans are to obtain client application identity from the 2-Factor Authentication (2FA) token when fully implemented.
DUZ("TEST")	This variable is used during menu generation. It indicates to the user when they are in a Test account by inserting the phrase " <TEST ACCOUNT>" into the "Select…" main menu prompt. For example (see Figure 44): `Select VA FileMan <TEST ACCOUNT> Option:`

Figure 43: Displaying the DUZ Array for a Signed-on User at a Programmer Prompt



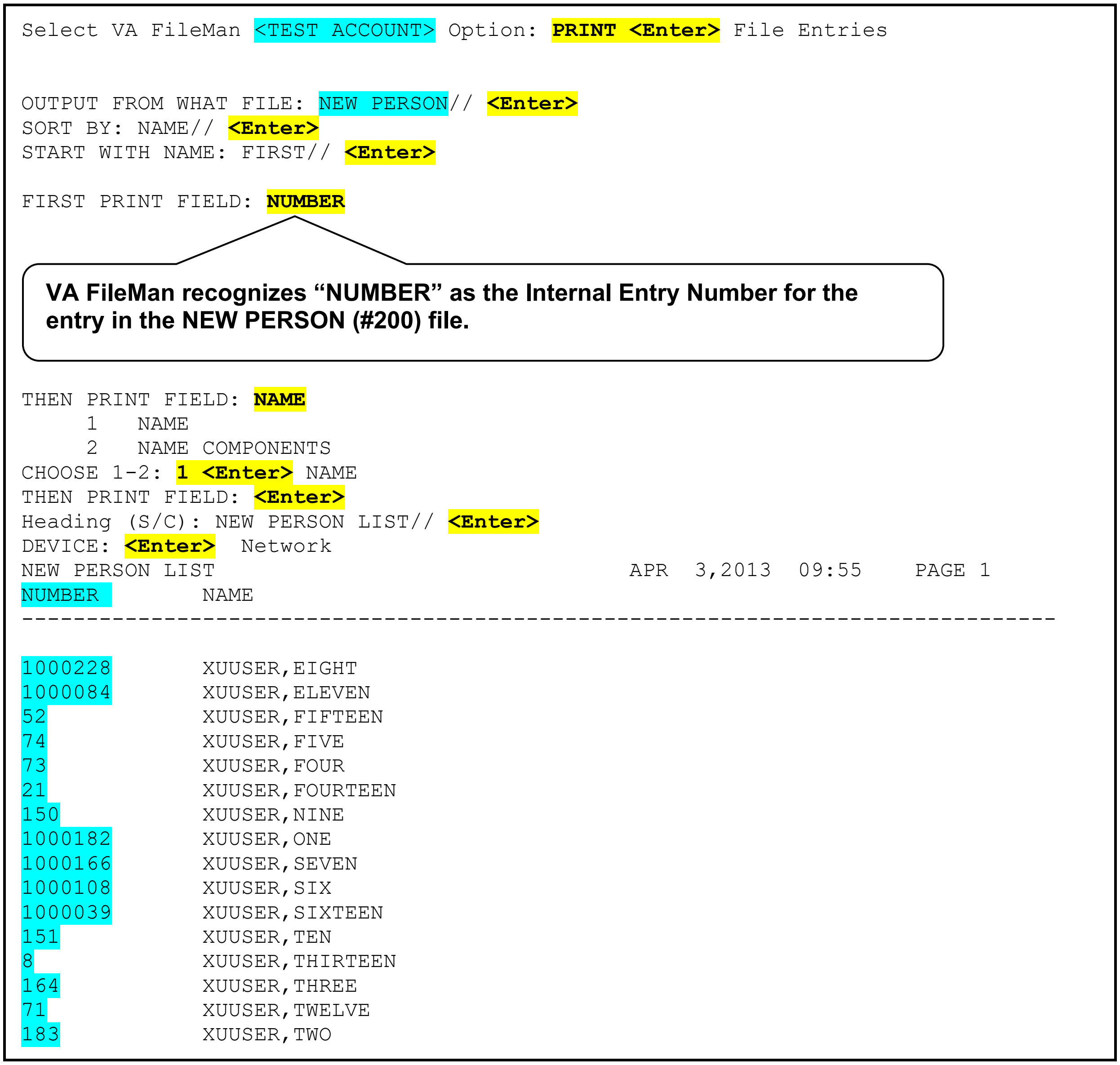
```
KRN>ZW DUZ
```

This Internal Entry Number (IEN) is always a unique number for each user entry in the NEW PERSON (#200) file.

```
DUZ=8
DUZ(0)=“@”
DUZ(1)=“”
DUZ(2)=2
DUZ(“AG”)=“V”
DUZ(“AUTO”)=1
DUZ(“BUF”)=1
DUZ(“LANG”)=1
DUZ(“TEST”)=“ <TEST ACCOUNT>”
```

When you want to display/print the **DUZ**, VA FileMan recognizes that when you enter “NUMBER” as a print field that you want to display/print the **DUZ** for the user entry from the NEW PERSON (#200) file.

Figure 44: Displaying the DUZ (Internal Entry Number) in a VA FileMan Report



```
Select VA FileMan <TEST ACCOUNT> Option: PRINT <Enter> File Entries


OUTPUT FROM WHAT FILE: NEW PERSON// <Enter>
SORT BY: NAME// <Enter>
START WITH NAME: FIRST// <Enter>

FIRST PRINT FIELD: NUMBER
```

VA FileMan recognizes “NUMBER” as the Internal Entry Number for the entry in the NEW PERSON (#200) file.

```
THEN PRINT FIELD: NAME
     1   NAME
     2   NAME COMPONENTS
CHOOSE 1-2: 1 <Enter> NAME
THEN PRINT FIELD: <Enter>
Heading (S/C): NEW PERSON LIST// <Enter>
DEVICE: <Enter>  Network
NEW PERSON LIST                                APR  3,2013  09:55    PAGE 1
NUMBER        NAME
--------------------------------------------------------------------------------


1000228       XUUSER,EIGHT
1000084       XUUSER,ELEVEN
52            XUUSER,FIFTEEN
74            XUUSER,FIVE
73            XUUSER,FOUR
21            XUUSER,FOURTEEN
150           XUUSER,NINE
1000182       XUUSER,ONE
1000166       XUUSER,SEVEN
1000108       XUUSER,SIX
1000039       XUUSER,SIXTEEN
151           XUUSER,TEN
8             XUUSER,THIRTEEN
164           XUUSER,THREE
71            XUUSER,TWELVE
183           XUUSER,TWO
```

4.2.8.2 Using Ranges of File Numbers

Can files be specified by number ranges? Yes; it is useful to do this when granting several files at once. First, find out the number of the files. Typing a question mark (**?**) at the “to Files:” prompt displays the number and name of the files. Note the numbers and then put them together on one line. You can use hyphens to indicate a consecutive range and commas to separate the single numbers and hyphenated groups as follows:

2,3,4,6,7,8,125,236,799

OR

2-4,6-8,125,236,799

File numbers are also used when printing a group of consecutive files. The prompt asks for a place to start with a default file name presented. To print just this one file, respond to the next prompt by simply pressing the **<Enter>** key, thereby accepting the default of ending after printing that one file.

To print a consecutive range of files, the lowest number is entered as the starting point and the highest number as the ending point. All files that fall in this range are printed.

4.2.8.3 Queuing File Access Specifications

Most of the options provide the opportunity to queue, after specifying who is to be granted which files. Queuing sends the specifications to TaskMan to assign to users at a later time. TaskMan can work at an off-peak time (e.g., midnight) to avoid consuming system resources during the daytime. If the system is *not* busy, queuing is still a good idea since your terminal is otherwise tied up while the report is being printed.

4.3 Running the File Access Security Conversion

4.3.1 Advantages

To implement File Access Security you need to run a conversion. Some advantages of implementing File Access Security include:

- **Easier to identify levels of access—**Running the conversion makes it possible to identify the levels of access each individual user has to each file.
- **Enhanced system performance—**Checking file access by user is slightly faster in terms of global accesses and CPU time.

4.3.2 Advance Preparation for the Conversion

The File Access Security conversion is designed to allocate access privileges to all of your users according to their current FILE MANAGER ACCESS CODE (#3) field value in the NEW PERSON (#200) file, **DUZ(0)**, combined with information about their file access through options stored in the **^DISV** global. After the conversion you should get only a few user requests for file access. The **File Access Security** [XUFILEACCESS] menu, an option on the **User Management** [XUSER] menu, should then be used to add a file to a user's ACCESSIBLE FILE (#32) Multiple field in the NEW PERSON (#200) file.

The conversion uses the FILE MANAGER ACCESS CODE (#3) field [**DUZ(0)** string] to assign file access according to the characters in the string. If a file is protected with a particular character that matches one in the user's code, that file is entered into the user's ACCESSIBLE FILE (#32) Multiple field. Levels of access are granted according to the file's original security (field-level security continues to function the same, by checking the FILE MANAGER ACCESS CODE (#3) field).

NOTE: Users with Programmer-level access (FILE MANAGER ACCESS CODE [#3] field = **@**) does *not* need to have any files in their ACCESSIBLE FILE (#32) Multiple field, since they are able to access *all* files *without* restriction.

4.3.2.1 ^DISV Global

The File Access Security conversion process makes use of the **^DISV** global to identify which files have recently been accessed by which users. The conversion adds all files that the user has been able to access (select from) to the user's ACCESSIBLE FILE (#32) Multiple field list. It grants **READ** access to these files.

Using the **^DISV** global to grant file access has the benefit of permitting option usage "as usual" the day after the conversion is run. **KILL**ing the **^DISV** global just *before* the conversion is *not* advised, since many users suffer inappropriate access restrictions and need special attention by system administrators just after the conversion. **KILL**ing the **^DISV** global a week or two before the conversion, however, may be worthwhile as a way of purging obsolete user data. In multi-CPU environments, where each CPU has its own copy of the **^DISV** global, you should choose the busiest user node upon which to run the conversion (in order to pick up the most comprehensive information from that node's **^DISV**). Caché sites should run the conversion from their busiest user node.

It is assumed that **^DISV** is *not* translated, so **K ^DISV** on the CPU where the conversion is run. Do this about two weeks before you perform the conversion, as advance preparation. **^DISV** is reset as soon as a user responds to a "Select:" prompt.

Figure 45: KILLing ^DISV—Sample Code

```
>K ^DISV
```

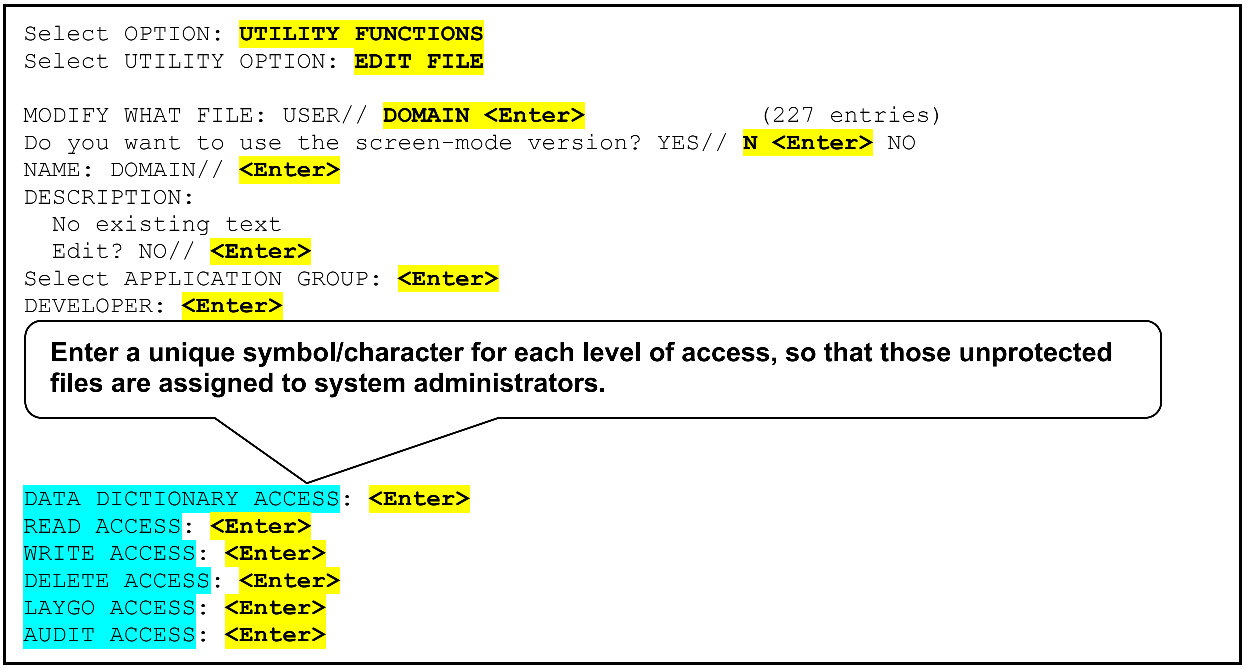
Do this only on the CPU where the conversion will run, about two weeks beforehand, as advance preparation.

4.3.2.2 Adding Explicit File Access for System Administrators

If there are any files that are neither protected nor accessed by users (e.g., the DOMAIN [#4.2] file) the conversion does *not* list them in any user's ACCESSIBLE FILE (#32) Multiple field. Before the conversion, these types of files are accessible to everyone, while after the conversion these files are only accessible to users with programmer-level access. Therefore, before the conversion, assign a unique symbol/character to otherwise unprotected files. This ensures that at least those users with that unique symbol (e.g., system administrators) are granted access. VA FileMan's **Edit File** [DIEDFILE] option can be used to edit the codes.

NOTE: In previous documentation and data dictionaries, it has been *implied* that the pound sign ("#") symbol/character was reserved for File Access Security for system administrators; however, this is *not* true. It has merely been used as a *convention.*

Figure 46: Updating File Access Settings (*Before* Conversion)



```
Select OPTION: UTILITY FUNCTIONS
Select UTILITY OPTION: EDIT FILE

MODIFY WHAT FILE: USER// DOMAIN <Enter>            (227 entries)
Do you want to use the screen-mode version? YES// N <Enter> NO
NAME: DOMAIN// <Enter>
DESCRIPTION:
  No existing text
  Edit? NO// <Enter>
Select APPLICATION GROUP: <Enter>
DEVELOPER: <Enter>
```

Enter a unique symbol/character for each level of access, so that those unprotected files are assigned to system administrators.

```
DATA DICTIONARY ACCESS: <Enter>
READ ACCESS: <Enter>
WRITE ACCESS: <Enter>
DELETE ACCESS: <Enter>
LAYGO ACCESS: <Enter>
AUDIT ACCESS: <Enter>
```

4.3.3 Summary of the File Access Security Conversion

The File Access Security conversion prepares the NEW PERSON (#200) file for VA FileMan's method of file access (lookup into a user's record for file access). VA FileMan's ability to protect data within files on fields and templates remains the same. The summary steps that occur when the conversion is run are outlined below:

1. Setup structure. The structure for implementing the file access method is set up via the following:
    a. Place the data dictionary (DD) for the ACCESSIBLE FILE (#32) Multiple field in the NEW PERSON (#200) file. This multiple is permanently put in place by running the File Access Security conversion.
    b. Install menu options, help frames, and templates used for maintaining the user file access method (i.e., entries with the **XUFI** namespace).

2. Add protected files to the ACCESSIBLE FILE (#32) Multiple field. Each user's FILE MANAGER ACCESS CODE (#3) field is used to add entries to the ACCESSIBLE FILE (#32) Multiple field as follows:
    a. Create a list of files to be processed by examining each file's protection codes. Files that meet *both* of the following requirements are temporarily stored in the **^UTILITY($J** global:
        - Files that have protection defined.
        - Files with protection *not* equal to **@**.

        NOTE: Files that lack any protection are bypassed. Such unprotected files are *not* later listed in anyone's ACCESSIBLE FILE (#32) Multiple field. Protection should therefore be applied *before* running the conversion so that at least some users (e.g., system administrators) are granted access.

    b. Examine each user in the NEW PERSON (#200) file. Each user meeting *all* of the following requirements is selected for further processing:
        - Users *not* terminated.
        - Users with an Access code.
        - Users with a VA FileMan Access code (i.e., FILE MANAGER ACCESS CODE [#3] field in the NEW PERSON [#200] file).
        - Users with a FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON [#200] file *not* equal to **@**.

        The user's FILE MANAGER ACCESS CODE (#3) field in the NEW PERSON [#200] file is parsed. Each symbol/character is compared with the list of files in the

^UTILITY($J global. All files that have a protection code matching this symbol/character are added to the user's ACCESSIBLE FILE (#32) Multiple field in the NEW PERSON [#200] file. If the symbol/character is used as the file's DATA DICTIONARY ("DD") file security, the user is granted **DD** access; if it is used as LAYGO, the user is granted **LAYGO** access, and so on.

3. Add files accessed by the user to the ACCESSIBLE FILE (#32) Multiple field. Files accessed by the user through options since the last time the **^DISV** global was **KILL**ed are added to the user's ACCESSIBLE FILE (#32) Multiple field by the processing of the **^DISV** global. Entries in **^DISV** that meet *both* of the following requirements are added to the ACCESSIBLE FILE (#32) Multiple field, with **READ** access:

    - The file *must not* be in VA FileMan's file number range (i.e., file number *must* be equal to or greater than **1.1**.
    - The user does *not* already have access to this file.

4.3.4 File Access Security Conversion Instructions

The steps that occur when the file access security conversion is run are described below:

1. Identify unprotected files and assign protection codes as desired (as described in the "Advance Preparation for the Conversion" section). For example, the DOMAIN (#4.2) file may need to be protected so that it is granted to users having a FILE MANAGER ACCESS CODE (#3) field containing the assigned symbol/character.

    NOTE: In previous documentation and data dictionaries, it has been *implied* that the pound sign ("**#**") symbol/character was reserved for File Access Security for system administrators; however, this is *not* true. It has merely been used as a *convention.*

2. Review the FILE MANAGER ACCESS CODE fields (#3) of VA FileMan users. The codes should contain symbols/characters matching those used to protect the files that these individuals use. Since the conversion automatically grants files to users according to previous privileges as indicated by the FILE MANAGER ACCESS CODE (#3) field, add any additional symbols/characters to their FILE MANAGER ACCESS CODE fields (#3) to take advantage of the conversion's automated file assignment according to levels of access.

3. Be ready to use the **File Access Security** [XUFILEACCESS] menu, Figure 43, to review and grant file access privileges *after* the conversion.

4. In the Production account, enable File Access Security system features and options with ENABLE^XUFILE3, as illustrated in Figure 47:

Figure 47: Enabling File Access Security—Sample User Dialog

```
In VAH:

>D ENABLE^XUFILE3

>
```

5. In the Production account, begin the conversion with ^XUINCON:

Figure 48: ^XUINCON Conversion Routine—Sample User Dialog

```
In VAH:

>D ^XUINCON

Version 7 of the Kernel defined a new multiple-valued field
in the New Person File called Accessible File.  This conversion
will store file access in this multiple in the following manner:

Those Users who have a FileMan Access Code (DUZ(0)) which
is not null, i.e., contains some character string,
will have their access string matched to the protection
currently on your files.  For each match between the file
and the user, the file will be listed in the user’s
Accessible File multiple as will the type of access
(dictionary, delete, laygo, read, write, audit).

NOTE: Files with no protection will NOT be assigned to any user.

Would you like to run the conversion now? NO//
```

6. If you are ready to run the conversion, answer **YES**:

Figure 49: Running a Conversion—Sample User Dialog

```
Would you like to run the conversion now? NO// YES
56237,36565
Build Table.
Convert Users.
Give access from DISV file.
X-ref.
Done56237,36565.
>
```

7. Review the newly assigned access settings. Use the **File Access Security** [XUFILEACCESS] menu, Figure 43, located on the **User Management** [XUSER] menu, to display file access by user and by file.

4.3.5 After the File Access Security Conversion

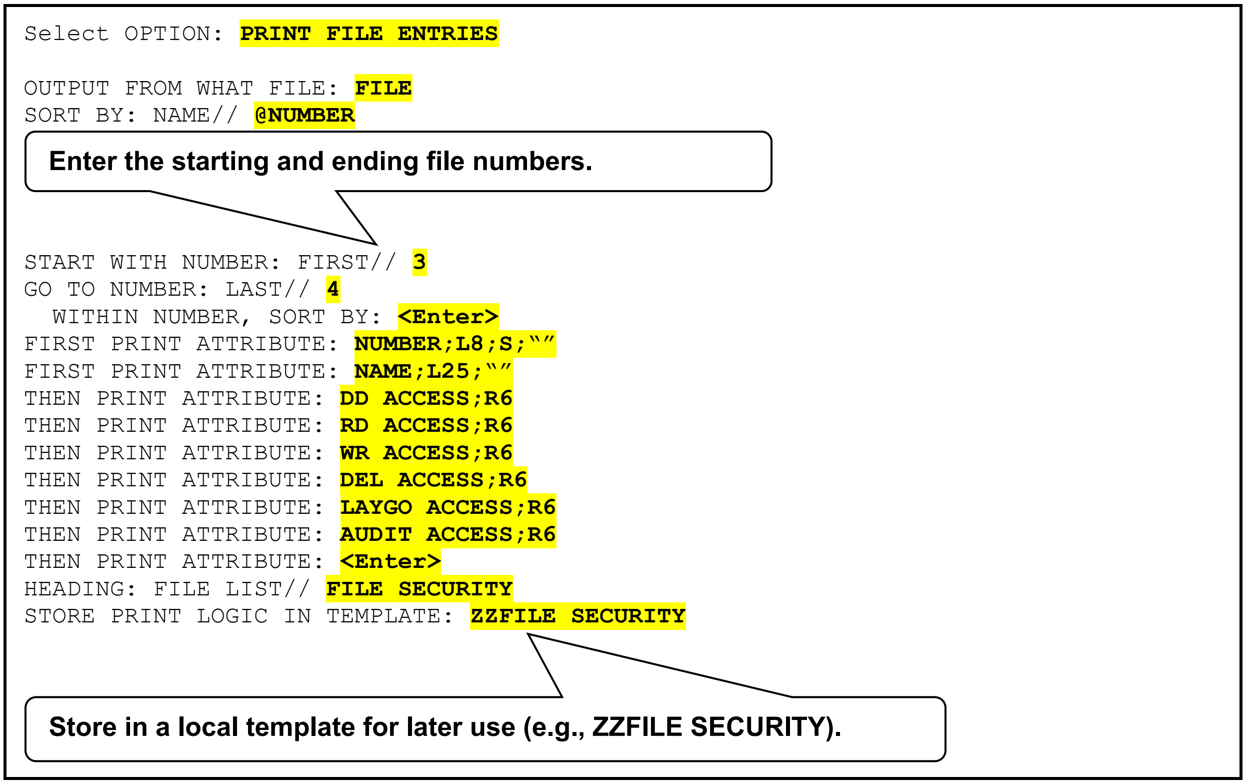
After the file access security conversion, users may complain about *not* being able to add entries to files as they previously could. This typically results from use of an option that navigates from one file to another. To be able to add entries to the navigated-to file, the user needs **LAYGO** access to that file. System administrators can solve the problem by granting **LAYGO** access using the **File Access Security** [XUFILEACCESS] menu options, Figure 43.

If this form of security is implemented, system administrators should find that it provides a more accurate and precise knowledge of who has what level of access to which files. When the conversion is run, privileges are granted to existing users by making use of information stored in the VA FileMan record of file manipulation activity, the **^DISV** global. The file access conversion grants each user **READ** access to files that the user had recently accessed as indicated in the **^DISV** global. System administrators can grant file access privileges to new users by copying the profile of an existing user with similar duties (e.g., a laboratory application coordinator or admissions clerk).

To be sure that appropriate levels of access have been allocated, system administrators should determine who has what level of access to which files. Access to sensitive files (e.g., the NEW PERSON [#200] file) should be reviewed and readjusted for individual users as appropriate. All files on a system should be reviewed before and after running the File Access Security conversion.

Figure 50 shows how to create a PRINT template to display a report on the current file access security:

Figure 50: Creating a PRINT Template to Display File Access Security—Sample User Dialog



```
Select OPTION: PRINT FILE ENTRIES

OUTPUT FROM WHAT FILE: FILE
SORT BY: NAME// @NUMBER
```

Enter the starting and ending file numbers.

```
START WITH NUMBER: FIRST// 3
GO TO NUMBER: LAST// 4
  WITHIN NUMBER, SORT BY: <Enter>
FIRST PRINT ATTRIBUTE: NUMBER;L8;S;“”
FIRST PRINT ATTRIBUTE: NAME;L25;“”
THEN PRINT ATTRIBUTE: DD ACCESS;R6
THEN PRINT ATTRIBUTE: RD ACCESS;R6
THEN PRINT ATTRIBUTE: WR ACCESS;R6
THEN PRINT ATTRIBUTE: DEL ACCESS;R6
THEN PRINT ATTRIBUTE: LAYGO ACCESS;R6
THEN PRINT ATTRIBUTE: AUDIT ACCESS;R6
THEN PRINT ATTRIBUTE: <Enter>
HEADING: FILE LIST// FILE SECURITY
STORE PRINT LOGIC IN TEMPLATE: ZZFILE SECURITY
```

Store in a local template for later use (e.g., ZZFILE SECURITY).

Once the conversion has been run, you can use the **File Access Security** [XUFILEACCESS] menu, Figure 43, to print the accessible files for individual users. Thus, you can establish profiles that would be typical of groups of users (e.g., Nursing, Pharmacy, or other services). Then, when establishing an account for a new user or reactivating the access of a previously terminated user, the profile is available for copying to the new user.

5 Electronic Signatures

5.1 User Interface

An electronic signature is a security tool that software applications can use as an additional identification check. For example, software can require that an electronic signature be applied to a particular form or document before subsequent processing can continue.

Electronic signature codes are stored in the NEW PERSON (#200) file.

5.1.1 Electronic Signature code Edit Option

If you need to create an electronic signature for yourself, you can choose the **Electronic Signature code Edit** [XUSESIG] option, available from the **User's Toolbox** [XUSERTOOLS] menu.

You can enter a new electronic signature code or change an existing code. The length of the code *must* be between **6** and **20** uppercase characters. Requiring all uppercase allows the code to be verified with either uppercase or lowercase input, since lowercase is converted to uppercase in the matching process. You should choose a code that other users are *not* likely to guess, as this code verifies that it is actually you who are signing off on some important action.

The **Electronic Signature code Edit** [XUSESIG] option also allows you to edit the following fields in the NEW PERSON (#200) file:

- INITIAL
- SIGNATURE BLOCK PRINTED NAME (#20.2)
- SIGNATURE BLOCK TITLE (#20.3)
- OFFICE PHONE (#.132)
- VOICE PAGER (#.137)
- DIGITAL PAGER (#.138)

Applications can print some or all of these fields when printing an electronically signed document. You should therefore ensure that the values entered in these fields are accurate.

5.1.1.1 Electronic Signature Code Edit Restrictions

As of Patch XU*8.0*679, the system restricts entries in the following fields in the NEW PERSON (#200) file when accessed through the **Electronic Signature code Edit** [XUSESIG] option:

- SIGNATURE BLOCK PRINTED NAME (#20.2)
- SIGNATURE BLOCK TITLE (#20.3)

As noted in Section 5.1.1, you can use the **Electronic Signature code Edit** [XUSESIG] option to enter and maintain electronic signature information for yourself. However, if the patch restrictions are activated, then access to the SIGNATURE BLOCK PRINTED NAME (#20.2) and SIGNATURE BLOCK TITLE (#20.3) fields will be enabled only for users who have the XUSIG security key assigned.

To enable restrictions, authorized site personnel *must* set the XU SIG BLOCK DISABLE general parameter to a value of **ON** (**1**). The parameter definition is stored in the PARAMETER DEFINITION (#8989.51) file, and the parameter data is stored in the PARAMETER (#8989.5) file:

- If the XU SIG BLOCK DISABLE parameter is set to **ON** and the user has the XUSIG security key assigned in the NEW PERSON (#200) file, then access to the restricted fields is allowed.
- If the XU SIG BLOCK DISABLE parameter is set to **ON**, but the user does *not* have the XUSIG security key assigned, then access to the restricted fields is *not* allowed.
- If the site leaves the XU SIG BLOCK DISABLE parameter set to **OFF** (**0**), then access to all electronic signature fields is allowed.

To set the XU SIG BLOCK DISABLE parameter to **ON**:

1. Log into VistA with programmer access.
2. At the “Select OPTION NAME:” prompt, enter **XPAR MENU TOOLS**.
3. At the “Select General Parameter Tools Option:” prompt, enter **EP**.
4. At the “Select PARAMETER DEFINITION NAME:” prompt, enter **XU SIG BLOCK DISABLE**.
5. At the “Sig Block Disable:” prompt, enter **YES**.

5.2 System Management

Figure 51: User Edit Menu Options

```
SYSTEMS MANAGER MENU ...                                                [EVE]
User Edit ...                                                         [XUSER]
    Electronic Signature Block Edit                           [XUSESIG BLOCK]
    Clear Electronic signature code  <locked:  XUMGR>         [XUSESIG CLEAR]
```

5.2.1 Electronic Signature Block Edit Option

The **Electronic Signature Block Edit** [XUSESIG BLOCK] option lets you edit the electronic signature code for any user on the system. When you create an electronic signature code for a user, the SIGNATURE BLOCK PRINTED NAME field in the NEW PERSON (#200) file is initially filled in by a cross-reference on the NAME (#.01) field (and is overwritten if the NAME [#.01] field is changed). Credentials (e.g., “**M.D.**”) can be added to customize the printed name. As a security feature, an input transform requires that the user’s last name (first comma piece of the NAME (#.01) field) be included in the printed name. (This field *cannot* be edited through VA FileMan, since it is **WRITE**-protected with a caret [^].)

5.2.1.1 Electronic Signature Block Edit Restrictions

As of Patch XU*8.0*679, the system restricts entries in the following fields in the NEW PERSON (#200) file when accessed through the **Electronic Signature Block Edit** [XUSESIG BLOCK] option:

- SIGNATURE BLOCK PRINTED NAME (#20.2)
- DEGREE (#10.6)

As noted in Section 5.2.1, you can use the **Electronic Signature Block Edit** [XUSESIG BLOCK] option to enter and maintain electronic signature information for other users. However, if the patch restrictions are activated, access to the SIGNATURE BLOCK PRINTED NAME (#20.2) and DEGREE (#10.6) fields will be enabled only for users who have the XUSIG security key assigned. To enable supervisors who hold the XUSIG security key to change the SIGNATURE BLOCK TITLE (#20.3) field for other users when patch restrictions are in force, Patch XU*8.0*679 added access to that field through the **Electronic Signature Block Edit** [XUSESIG BLOCK] option.

REF: For instructions on activating restrictions, see Section 5.1.1.1, Electronic Signature Code Edit Restrictions.

To maintain valid educational credentials, DEGREE (#10.6) field entries made through the **Electronic Signature Block Edit** [XUSESIG BLOCK] option are restricted to valid degrees stored in the EDUCATION (#20.11) file. To support this functionality, Patch XU*8.0*679 added the **EDUCATION (Degree) File Edit** [XUSESIG DEG] option to the **User Management** [XUSER] menu. The **EDUCATION (Degree) File Edit** [XUSESIG DEG] option, which is locked by the XUSIG security key, enables maintenance of DEGREE field entries in the EDUCATION (#20.11) file. VistA uses these records to validate user entries when appending one or more degrees to an electronic signature. This validation applies even when the XU SIG BLOCK DISABLE parameter is set to **OFF**.

NOTE: Because VistA automatically cross-references DEGREE (#10.6) field entries in the NEW PERSON (#200) file with the DEGREE (#6) field in the NAME COMPONENTS (#20) file, any updates made directly to the DEGREE (#6) field in the NAME COMPONENTS (#20) file will also be validated against degrees in the EDUCATION (#20.11) file.

5.2.2 Clear Electronic signature code Option

The **Clear Electronic signature code** [XUSESIG CLEAR] option is another option available to system administrators that allows the clearing (deleting) of an electronic signature code. This option is locked with the XUMGR security key. This option can be used to clear a user’s electronic signature code if the user has forgotten the code. The user can then enter a new code with the **Electronic Signature code Edit** [XUSESIG] option in the **User’s Toolbox** [XUSERTOOLS] menu.

6 DEA ePCS Utility

6.1 Overview

Kernel patch XU*8.0*580 was created in support of the Drug Enforcement Agency (DEA) e-Prescribing of Controlled Substances (ePCS) Utility using Public Key Infrastructure (PKI). This section describes the modifications and enhancements to Kernel (and other VistA software) to meet the requirements proposed by the DEA Interim Final Rule (IFR) for Electronic Prescriptions for Controlled Substances effective as of June 1, 2010.

NOTE: This document only describes the changes made to Kernel in support of the DEA ePCS Utility.

REF: For more information on the DEA ePCS Utility software and other VistA applications, see the following:

- Computerized Patient Record System (CPRS) documentation on the VDL: VDL CPRS Application Documents
- Pharmacy: Controlled Substances documentation: on the VDL: VDL Controlled Substances Application Documents

6.1.1 History

The Veterans Health Administration (VHA) Patient Care Services Office Pharmacy Benefits Management Services (PBM) requested enhancements to Veterans Health Information Systems and Technology Architecture (VistA), specifically the following software applications:

- Computerized Patient Record System (CPRS)
- Outpatient Pharmacy
- Controlled Substances
- Kernel

The enhancements made to these applications is to ensure that prescriptions for Controlled Substances (i.e., drugs listed in federal Controlled Substance Schedules II through V) can be digitally signed by the Prescribers and electronically transmitted from Prescribers to a Department of Veterans Affairs (VA) Pharmacy. The request was aimed at filling in the difference between the Hines Drug Enforcement Agency (DEA) ePrescribing pilot project as it stood as of April 2014 and the proposed DEA ePrescribing of Controlled Substances as shown in the June 27, 2008 Federal Register. These regulations allowed the process and proof of concept that was demonstrated with the DEA pilot to be expanded beyond the Hines VA Hospital facility.

The Hines VA/DEA Public Key Infrastructure (PKI) project stems from a pilot initiated in 2002 to demonstrate the ability for CPRS to incorporate digital signatures for Schedule II Controlled

Substance narcotic prescriptions. Hines VA Hospital was the pilot site and had previously been granted a waiver of regulations by the DEA to test the system.

The Pilot procedure was as follows:

1. Prescribers insert a “smart card” into a reader.
2. Prescribers enter an electronic prescription into CPRS.
3. System authenticates the Prescriber’s PKI prescribing credentials on the smart card.
4. System digitally signs the prescription.
5. System delivers the order to the VA pharmacy electronically.

The initial pilot evaluation, which allowed approximately 50 users to prescribe electronically using “smart cards”, was formally concluded in 2003. DEA authorized Hines VA Hospital to continue using the system in its current form until new regulations were published regarding electronic transmission of prescriptions using Personal Identity Verification (PIV) cards (aka smart cards). Subsequently, the VistA software was modified to meet the new standards.

Under the proposed DEA ePrescribing regulations, the CPRS system *must* authenticate the Prescriber’s credentials on a hard token (e.g., PIV card) and then display a mandatory message with DEA-required intent language that the Prescriber *must* consent to. Only after the Prescriber consents to the DEA-required wording can the prescription be transmitted to the VA Pharmacy.

The PIV card to be used for the DEA ePrescribing is the VA-wide PIV Card program mandated by Homeland Security Presidential Directive #12 (HSPD-12).

REF: For information on validating PIV cards, see the “PIV Card Validation—Revocation Server” section.

NOTE: CPRS requested the original funding of this software upgrade as part of the CPRS v29 funding submission.

6.1.2 Requirements

Once the DEA ePrescribing regulations were enacted, system changes were required to bring the VA in compliance with DEA regulations. The majority of the changes needed for the DEA ePCS Utility are in the VistA CPRS and Outpatient Pharmacy applications; however, there were also some changes needed in Kernel:

- CPRS—Allows VA Prescribers to enter and digitally sign prescriptions.
- Outpatient Pharmacy—Notifies a VA pharmacy that a prescription order was made in CPRS.
- Kernel—Provides the Application Programming Interfaces (APIs) between the VistA Pharmacy and CPRS applications that allow the PKI credentials on the smart card to be verified. The PIV technology ensures that the Prescriber’s credentials are vetted and

emplaced on the PIV card according to the DEA regulations once they are enacted into law.

The DEA regulations governing the electronic prescribing and transmission of Controlled Substances pertain to the following conditions:

- VA Prescribers of DEA-regulated Controlled Substances (Schedules II through V).
- Patients using a VA pharmacy.
- VA Pharmacists who fill the Controlled Substance prescriptions.
- Pharmacy Benefits Management (PBM), who has the accountability to minimize the abuse of Control Substances.

6.1.3 Benefits

The benefits of the DEA EPCS Utility include the following:

- Concise ordering of the correct prescriptions.
- Increased security against abuse of Controlled Substances—Test results showed a **90%** reduction in the number of forged, tampered or altered Controlled Substances presented to the pharmacy.
- An electronic record of prescription history that can be monitored and reported.
- Increased patient safety—Test results showed a **75%** reduction in the number of Controlled Substance prescription fill errors caused by illegible handwriting.
- Decreased wait time for patients to receive their prescriptions—Test results showed a **50%** reduction in the average time from when a prescription is written to when it is process (finished) by pharmacy, primarily affected by the elimination of prescription transit time from remote clinics.

6.1.4 Intended Audience

The intended audience of this manual is all key stakeholders. The stakeholders for the DEA ePCS Utility include the following:

- **(Primary) DEA-registered Prescribers of Controlled Substances—**Users who do the following:
    - Create the prescription order in the system.
    - Digitally sign the prescription.
    - Submit the prescription electronically to the Pharmacy.

Under the proposed DEA regulations, these users also electronically reject or agree to DEA-mandated wording prior to electronically signing the prescription.

- **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. These users are also responsible for the following:
    - Installing the necessary hardware and software for use of the smart card-based digital certificates.
    - Maintaining the server that runs the Certificate Revocation List (CRL) and other signature-checking processes.
    - Assisting in the maintenance of the database containing all valid DEA registrants within the VA. This database is an entity outside of VistA. The management of this database is shared by the VA and DEA.

- **Information Security Officer (ISO)—**The ISO is responsible for information security at each VA site.
- **Emerging Health Technologies (EHT)—**Users who identify, explore, pilot, and move into Production those technologies that can contribute to VA business needs. In this instance, the Public Key Infrastructure (PKI) technologies.
- **Personal Identification Verification (PIV) Project—**This VA project provides formatted smart cards for use with the system. The PIV project personnel ensure that the DEA PKI expansion for digitally signing and transmitting electronic prescriptions fits in with the scope and objectives of the Veterans Health Administration (VHA)-wide Homeland Security Presidential Directive (HSPD)-12 mandated directives.
- **Drug Enforcement Agency (DEA)—**The Federal agency that:
    - Enforces the Controlled Substances laws and regulations of the United States.
    - Enforces provisions of the Controlled Substances Act as they pertain to the manufacture, distribution, and dispensing of legally produced Controlled Substances.
    - Assists in the maintenance of the database containing all valid DEA registrants within the VA. This database is an entity outside of VistA. The management of this database is shared by the VA and DEA.

- **Office of Information and Technology (OIT)—**VistA legacy development teams.
- **Product Support (PS)**.

6.2 Processes

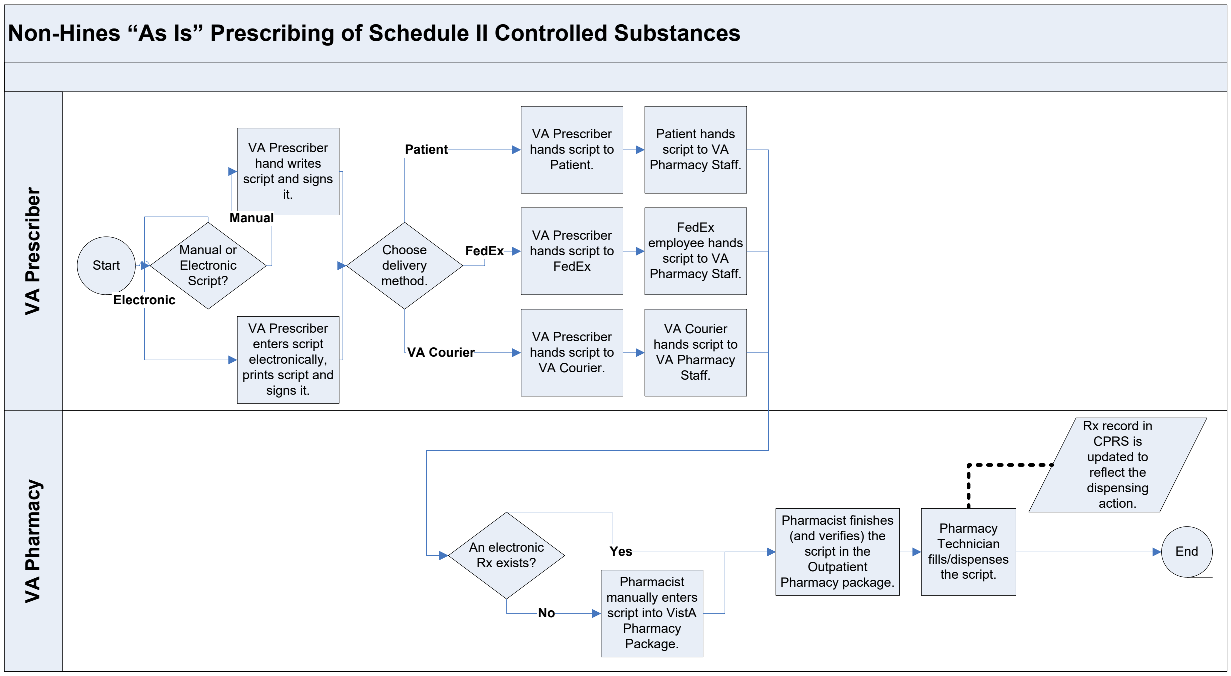
6.2.1 Manual Paper-based Process

For Schedule II Controlled Substance prescriptions within the VA using the manual paper-based process, the procedure is as follows:

1. VA Prescriber either hand-writes a prescription before signing it or prints off a prescription form and hand-signs it before giving it to the patient.
2. Patient or courier then hand-delivers the paper prescription form to the VA pharmacist.
3. VA Pharmacist manually enters the script into the VistA Pharmacy package.
4. After filling the prescription, the VistA Outpatient Pharmacy package updates CPRS with the record of the new fill.

With this method, CPRS has no way to verify the credentials of the Prescriber when a prescription order is hand written. Additionally, when the hand-written script is illegible, the VA Pharmacist either guesses at what the Prescriber intends or *must* call the Prescriber to ascertain what the Prescriber intended on the handwritten script. In either of these cases, the prescription fill is delayed, and the VA patient *must* wait for their medically necessary medication.

Figure 52: DEA ePCS—Manual Paper-based Process to Prescribe Schedule II Controlled Substances

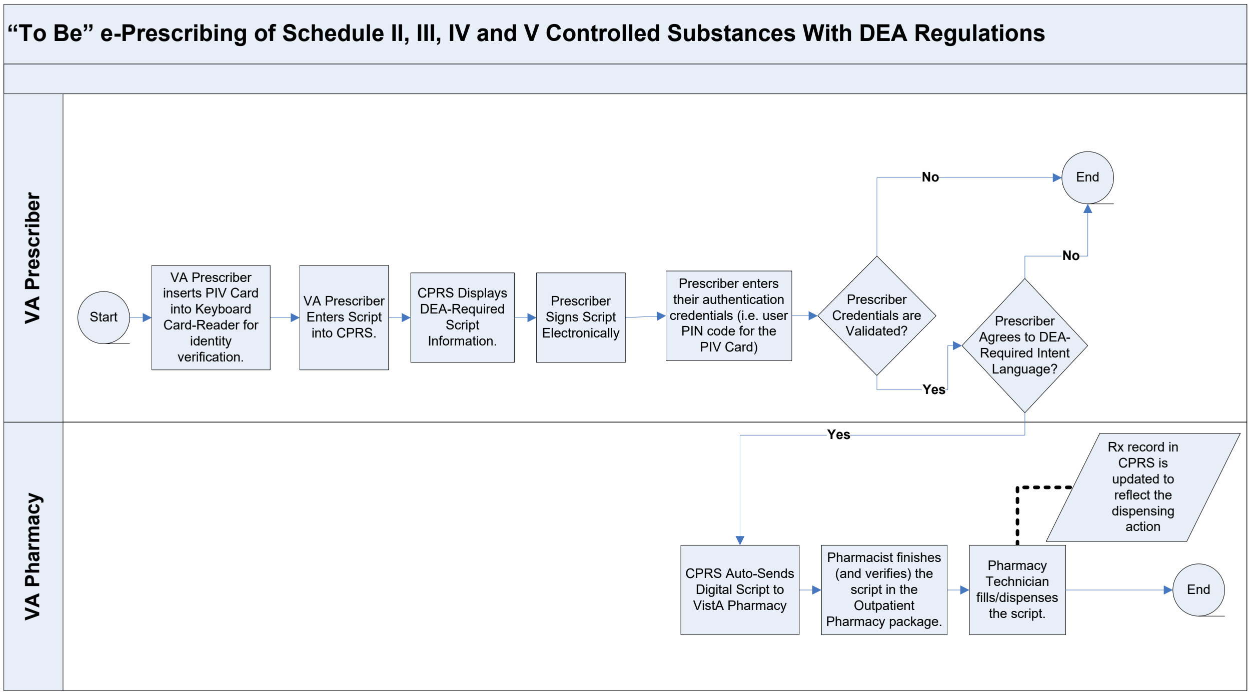

Non-Hines "As Is" Prescribing of Schedule II Controlled Substances
VA Prescriber
VA Pharmacy
Start
Manual or Electronic Script?
Manual
Electronic
VA Prescriber hand writes script and signs it.
VA Prescriber enters script electronically, prints script and signs it.
Choose delivery method.
Patient
FedEx
VA Courier
VA Prescriber hands script to Patient.
VA Prescriber hands script to FedEx
VA Prescriber hands script to VA Courier.
Patient hands script to VA Pharmacy Staff.
FedEx employee hands script to VA Pharmacy Staff.
VA Courier hands script to VA Pharmacy Staff.
An electronic Rx exists?
Yes
No
Pharmacist manually enters script into VistA Pharmacy Package.
Pharmacist finishes (and verifies) the script in the Outpatient Pharmacy package.
Pharmacy Technician fills/dispenses the script.
Rx record in CPRS is updated to reflect the dispensing action.
End

6.2.2 e-Prescribing Process

For Schedule II – V Controlled Substance prescriptions within the VA using the ePrescribing process (i.e., e-Prescribing of Controlled Substances [ePCS] Utility), the procedure is as follows:

1. VA Prescriber inserts a common access Personal Identity Verification (PIV) card (i.e., a smart card, which uniquely identifies the Prescriber) into a card reader attached to a computer keyboard.
2. VA Prescriber enters the prescription order into the Computerized Patient Record System (CPRS).
3. VA Prescriber signs the script electronically.
4. CPRS prompts the Prescriber to provide the credentials for the smart card (analogous to an Automated Teller Machine [ATM] card’s Personal Identification Number [PIN] code).
5. System verifies the PKI credentials.
6. System affixes a digital signature to the prescription (digitally signed).
7. CPRS sends the script order electronically to the VistA Pharmacy system.
8. VA Pharmacist fills the script in VistA Pharmacy.
9. VistA Pharmacy automatically sends a record of the prescription fill to CPRS.

Figure 53: DEA ePCS—ePrescribing Process to Prescribe Schedule II - V Controlled Substances


“To Be” e-Prescribing of Schedule II, III, IV and V Controlled Substances With DEA Regulations
VA Prescriber
VA Pharmacy
Start
VA Prescriber inserts PIV Card into Keyboard Card-Reader for identity verification.
VA Prescriber Enters Script into CPRS.
CPRS Displays DEA-Required Script Information.
Prescriber Signs Script Electronically
Prescriber enters their authentication credentials (i.e. user PIN code for the PIV Card)
Prescriber Credentials are Validated?
No
End
Yes
Prescriber Agrees to DEA-Required Intent Language?
No
Yes
CPRS Auto-Sends Digital Script to VistA Pharmacy
Pharmacist finishes (and verifies) the script in the Outpatient Pharmacy package.
Pharmacy Technician fills/dispenses the script.
Rx record in CPRS is updated to reflect the dispensing action
End

ℹ **REF:** For information on PIV and prescription validation processes, see the following sections:

- PIV Card Validation—Revocation Server
- Prescription Validation and Verification Process—PKIServer.exe Application

6.3 Configuring the DEA ePCS Utility

There are two steps to configure the DEA ePCS Utility:

1. Set the XUEPCS REPORT DEVICE Parameter.
2. Add DEA ePCS Utility Users.

6.3.1 Set the XUEPCS REPORT DEVICE Parameter

Set the XUEPCS REPORT DEVICE Parameter to the printer device. You can set this parameter by using either of the following methods:

- General Parameter Tools Menu.
- XPAREDIT Routine.

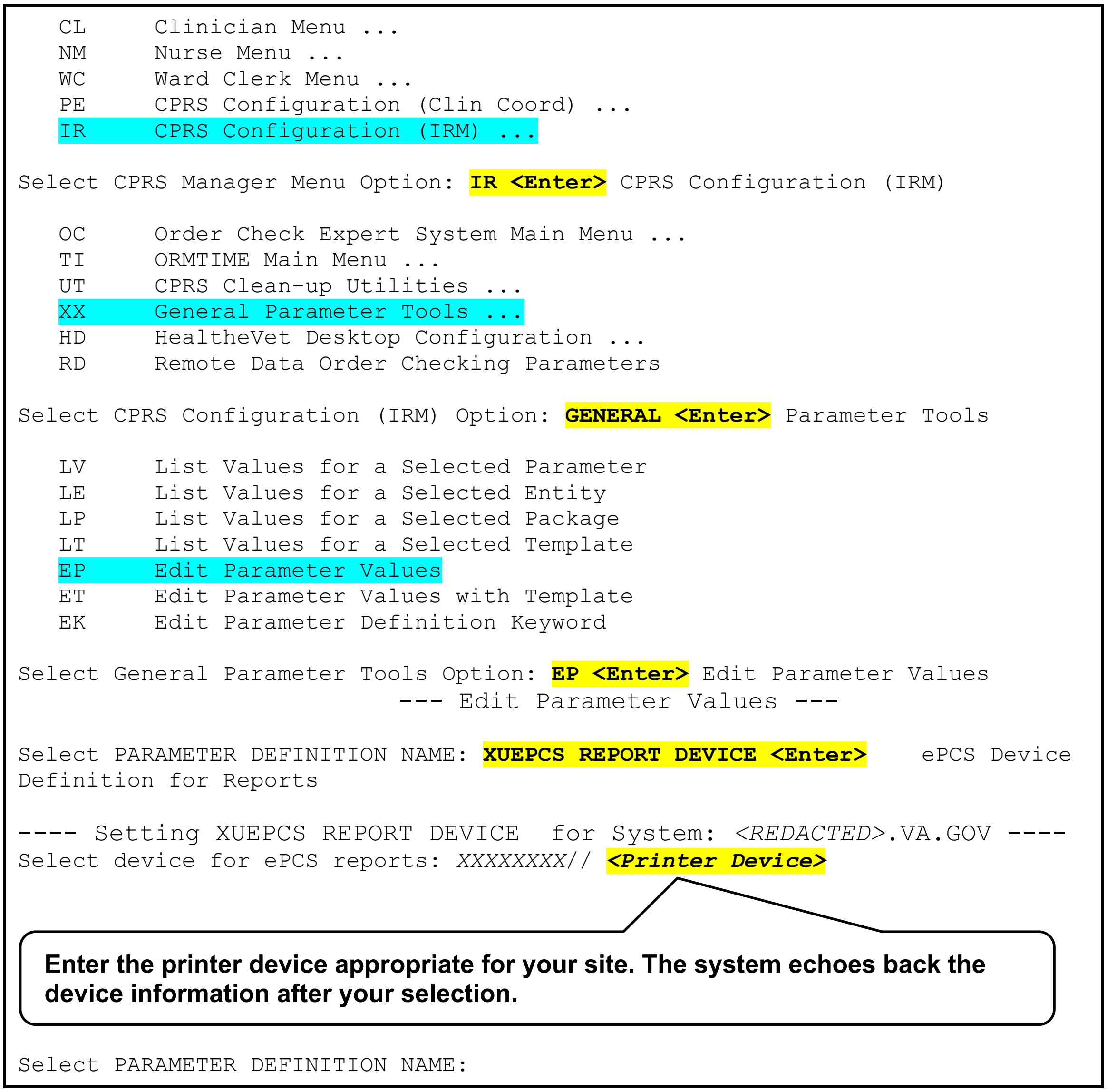
6.3.1.1 General Parameter Tools Menu

Use the **General Parameter Tools** [XPAR MENU TOOLS] menu, located under the **CPRS Configuration (IRM)** [OR PARAM IRM MENU] menu, to update the XUEPCS REPORT DEVICE parameter.

To edit the DEA ePCS Utility parameter, perform the following procedure:

1. From the **CPRS Manager Menu** [ORMGR], select the **IR—CPRS Configuration (IRM)** [OR PARAM IRM MENU] option.
2. At the “Select CPRS Configuration (IRM) Option:” prompt, select the **XX—General Parameter Tools** [XPAR MENU TOOLS]option.
3. At the “Select General Parameter Tools Option:” prompt, select the **EP—Edit Parameter Values** [XPAR EDIT PARAMETER] option.
4. At the “Select PARAMETER DEFINITION NAME:” prompt, enter **XUEPCS REPORT DEVICE**.
5. At the “Select device for ePCS reports: *XXXXXXXX*//” prompt, enter the printer device appropriate for your system.

Figure 54: DEA ePCS: General Parameter Tools Menu [XPAR MENU TOOLS]—Editing DEA ePCS Site Parameter



```
   CL     Clinician Menu ...
   NM     Nurse Menu ...
   WC     Ward Clerk Menu ...
   PE     CPRS Configuration (Clin Coord) ...
   IR     CPRS Configuration (IRM) ...

Select CPRS Manager Menu Option: IR <Enter> CPRS Configuration (IRM)

   OC     Order Check Expert System Main Menu ...
   TI     ORMTIME Main Menu ...
   UT     CPRS Clean-up Utilities ...
   XX     General Parameter Tools ...
   HD     HealtheVet Desktop Configuration ...
   RD     Remote Data Order Checking Parameters

Select CPRS Configuration (IRM) Option: GENERAL <Enter> Parameter Tools

   LV     List Values for a Selected Parameter
   LE     List Values for a Selected Entity
   LP     List Values for a Selected Package
   LT     List Values for a Selected Template
   EP     Edit Parameter Values
   ET     Edit Parameter Values with Template
   EK     Edit Parameter Definition Keyword

Select General Parameter Tools Option: EP <Enter> Edit Parameter Values
                            --- Edit Parameter Values ---

Select PARAMETER DEFINITION NAME: XUEPCS REPORT DEVICE <Enter>    ePCS Device
Definition for Reports

---- Setting XUEPCS REPORT DEVICE  for System: <REDACTED>.VA.GOV ----
Select device for ePCS reports: XXXXXXXX// <Printer Device>

Select PARAMETER DEFINITION NAME:
```

Enter the printer device appropriate for your site. The system echoes back the device information after your selection.

6.3.1.2 XPAREDIT Routine

Use the **XPAREDIT** routine to update the XUEPCS REPORT DEVICE parameter.

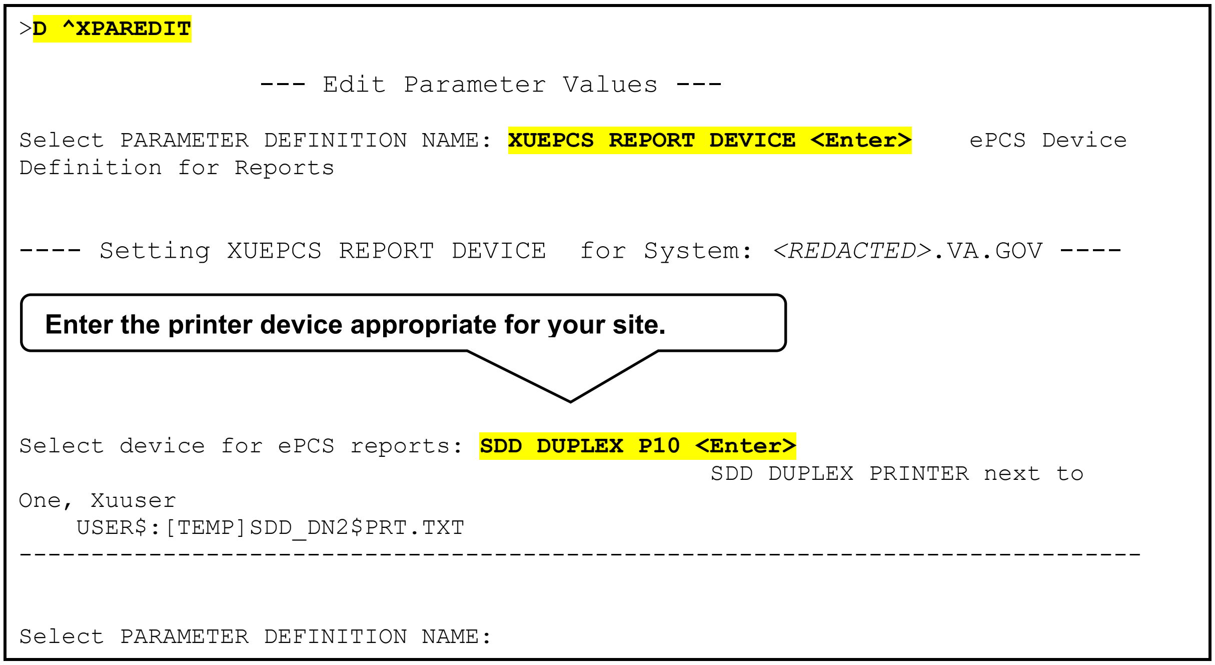
To edit the DEA ePCS Utility parameter, perform the following procedure:

1. From the programmer prompt, enter the following code:

   ```
   D ^XPAREDIT
   ```

2. At the “Select PARAMETER DEFINITION NAME:” prompt, enter **XUEPCS REPORT DEVICE**.
3. At the “Select device for ePCS reports: *XXXXXXXX*//” prompt, enter the printer or other device appropriate for your system.

Figure 55: DEA ePCS: XPAREDIT Routine—Editing DEA ePCS Site Parameter: Test Account



```
>D ^XPAREDIT

                --- Edit Parameter Values ---

Select PARAMETER DEFINITION NAME: XUEPCS REPORT DEVICE <Enter>    ePCS Device
Definition for Reports


---- Setting XUEPCS REPORT DEVICE  for System: <REDACTED>.VA.GOV ----
```

Enter the printer device appropriate for your site.

```
Select device for ePCS reports: SDD DUPLEX P10 <Enter>
                                                SDD DUPLEX PRINTER next to
One, Xuuser
    USER$:[TEMP]SDD_DN2$PRT.TXT
------------------------------------------------------------------------------


Select PARAMETER DEFINITION NAME:
```

6.3.2 Add DEA ePCS Utility Users

There are three steps to give a user access to the DEA ePCS Utility:

1. Assign the XUEPCSEDIT Security Key.
2. Assign the XU EPCS EDIT DATA Option.
3. Assign the XUSSPKI UPN SET Option.

6.3.2.1 Assign the XUEPCSEDIT Security Key

To assign the XUEPCSEDIT security key, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **Menu Management** [XUMAINT] menu.
2. At the “Select Menu Management Option:” prompt, select the **Key Management** [XUKEYMGMT] menu.
3. At the “Select Key Management Option:” prompt, select the **Allocation of Security Keys** [XUKEYALL] option.
4. At the “Allocate key:” prompt, enter **XUEPCSEDIT** security key.
5. At the “Another key:” prompt, press **Enter** to complete your entries.
6. At the “Holder of key:” prompt, enter the user’s name.
7. At the “Another holder:” prompt, enter any additional user names that need access to the DEA ePCS Utility. When complete, press **Enter**.

8. At the “You are allocating keys. Do you wish to proceed? YES//” prompt, press **Enter** to accept the **YES** default response.

Figure 56: DEA ePCS: Adding DEA ePCS Utility Users by Assigning the XUEPCSEDIT Security Key

```
Select Systems Manager Menu Option: MENU <Enter> Management


          Edit options
          Key Management ...
          Secure Menu Delegation ...
          Restrict Availability of Options
          Option Access By User
          List Options by Parents and Use
          Fix Option File Pointers
          Help Processor ...
   OPED   Screen-based Option Editor
          Display Menus and Options ...
          Edit a Protocol
          Menu Rebuild Menu ...
          Out-Of-Order Set Management ...
          See if a User Has Access to a Particular Option
          Show Users with a Selected primary Menu

Select Menu Management Option: KEY <Enter> Management


          Allocation of Security Keys
          De-allocation of Security Keys
          Enter/Edit of Security Keys
          All the Keys a User Needs
          Change user’s allocated keys to delegated keys
          Delegate keys
          Keys For a Given Menu Tree
          List users holding a certain key
          Remove delegated keys
          Show the keys of a particular user

Select Key Management Option: ALLOC <Enter> ation of Security Keys

Allocate key: XUEPCSEDIT

Another key: <Enter>

Holder of key: XUUSER,ONE <Enter>      OX          TECHNICAL WRITER

Another holder: <Enter>

You’ve selected the following keys:

XUEPCSEDIT

You’ve selected the following holders:

XUUSER,ONE

You are allocating keys.  Do you wish to proceed? YES// <Enter>

XUEPCSEDIT being assigned to:
     XUUSER,ONE
```

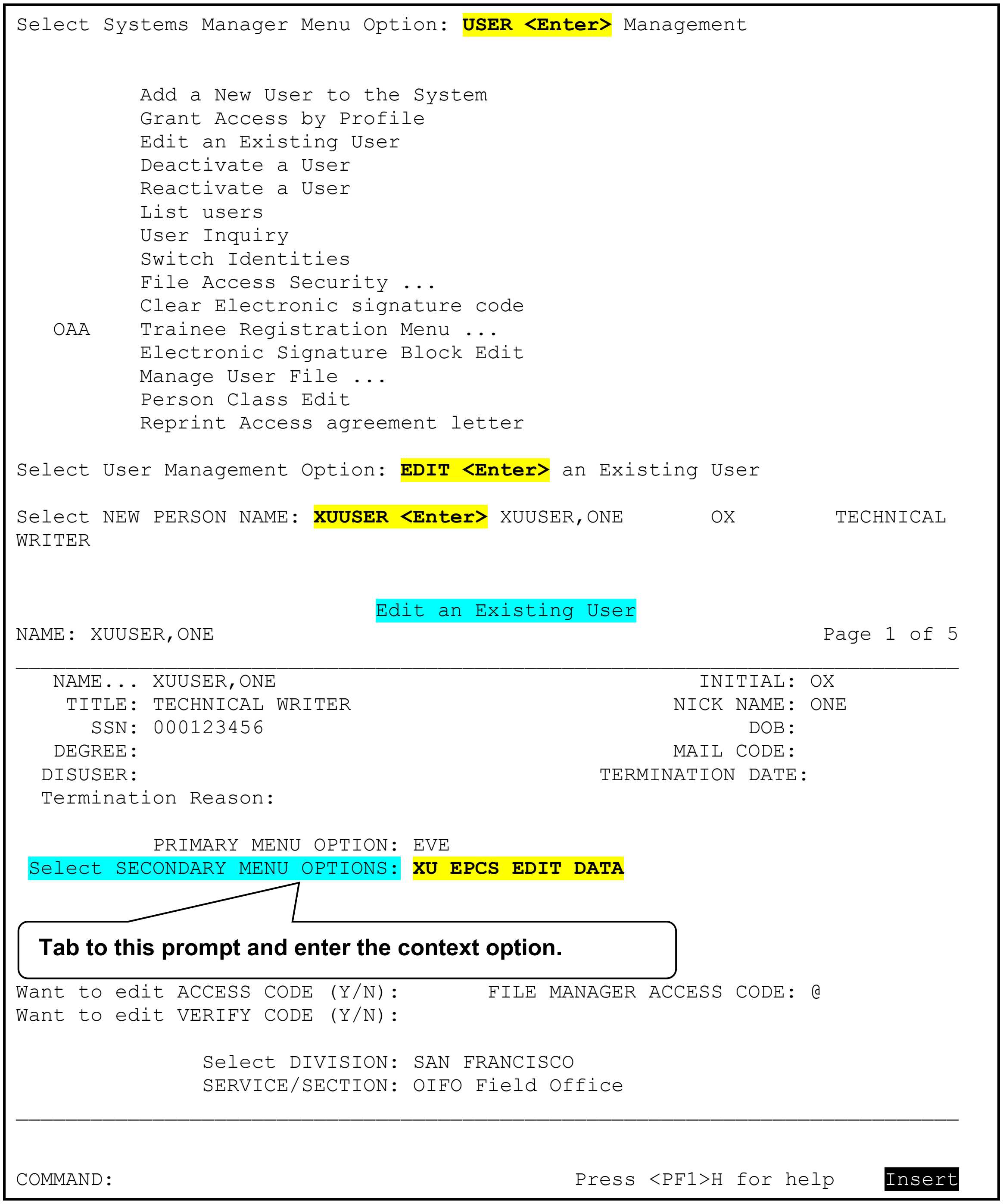
6.3.2.2 Assign the XU EPCS EDIT DATA Option

The **ePCS Edit Prescriber Data** [XU EPCS EDIT DATA] option is the context option the RPC Broker uses for the DEA ePCS Utility when making remote procedure calls.

To assign the **ePCS Edit Prescriber Data** [XU EPCS EDIT DATA] option for each user, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **User Management** [XUSER] menu.
2. At the “Select User Management Option:” prompt, select the **Edit an Existing User** [XUSEREDIT] option.
3. At the “Select NEW PERSON NAME:” prompt, enter the user’s name.
4. In the “Edit an Existing User” main screen, tab down to the “Select SECONDARY MENU OPTIONS:” prompt, enter the ePCS Edit Prescriber Data [XU EPCS EDIT DATA] option.
5. (Optional) In the “SECONDARY MENU OPTIONS” popup screen, tab to “SYNONYM:” prompt and enter a synonym for this context option.
6. Tab to the “COMMAND:” prompt, enter **Close**. The “SECONDARY MENU OPTIONS” popup screen closes.
7. Tab to the “COMMAND:” prompt, enter **Exit**. The “Edit an Existing User” main screen closes.

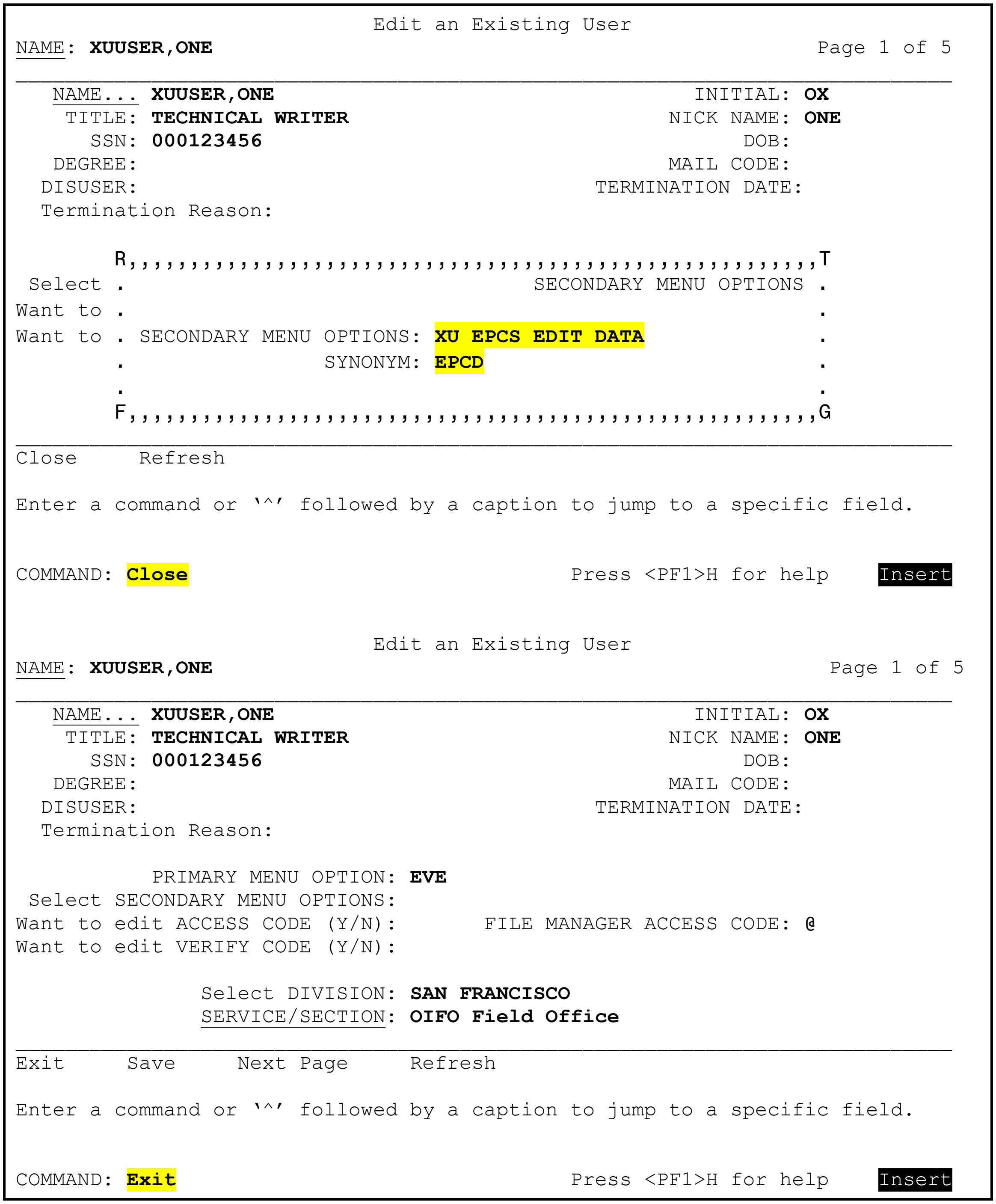
Figure 57: DEA ePCS: Assigning the XU EPCS EDIT DATA Option—Sample User Entries (1 of 2)



```
Select Systems Manager Menu Option: USER <Enter> Management


          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
   OAA    Trainee Registration Menu ...
          Electronic Signature Block Edit
          Manage User File ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management Option: EDIT <Enter> an Existing User

Select NEW PERSON NAME: XUUSER <Enter> XUUSER,ONE       OX        TECHNICAL
WRITER


                             Edit an Existing User
NAME: XUUSER,ONE                                                 Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                  INITIAL: OX
    TITLE: TECHNICAL WRITER                          NICK NAME: ONE
      SSN: 000123456                                       DOB:
   DEGREE:                                           MAIL CODE:
  DISUSER:                                     TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS: XU EPCS EDIT DATA


Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office
____________________________________________________________________________


COMMAND:                                     Press <PF1>H for help    Insert
```

Tab to this prompt and enter the context option.

Figure 58: DEA ePCS: Assigning the XU EPCS EDIT DATA Option—Sample User Entries (2 of 2)



```
                             Edit an Existing User
NAME: XUUSER,ONE                                                  Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                     INITIAL: OX
    TITLE: TECHNICAL WRITER                             NICK NAME: ONE
      SSN: 000123456                                          DOB:
   DEGREE:                                              MAIL CODE:
  DISUSER:                                       TERMINATION DATE:
  Termination Reason:

        R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
 Select .                                 SECONDARY MENU OPTIONS .
Want to .                                                        .
Want to . SECONDARY MENU OPTIONS: XU EPCS EDIT DATA              .
        .                SYNONYM: EPCD                           .
        .                                                        .
        F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
____________________________________________________________________________
Close     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: Close                                Press <PF1>H for help    Insert


                             Edit an Existing User
NAME: XUUSER,ONE                                                   Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                     INITIAL: OX
    TITLE: TECHNICAL WRITER                             NICK NAME: ONE
      SSN: 000123456                                          DOB:
   DEGREE:                                              MAIL CODE:
  DISUSER:                                       TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS:
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office
____________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: Exit                                 Press <PF1>H for help    Insert
```

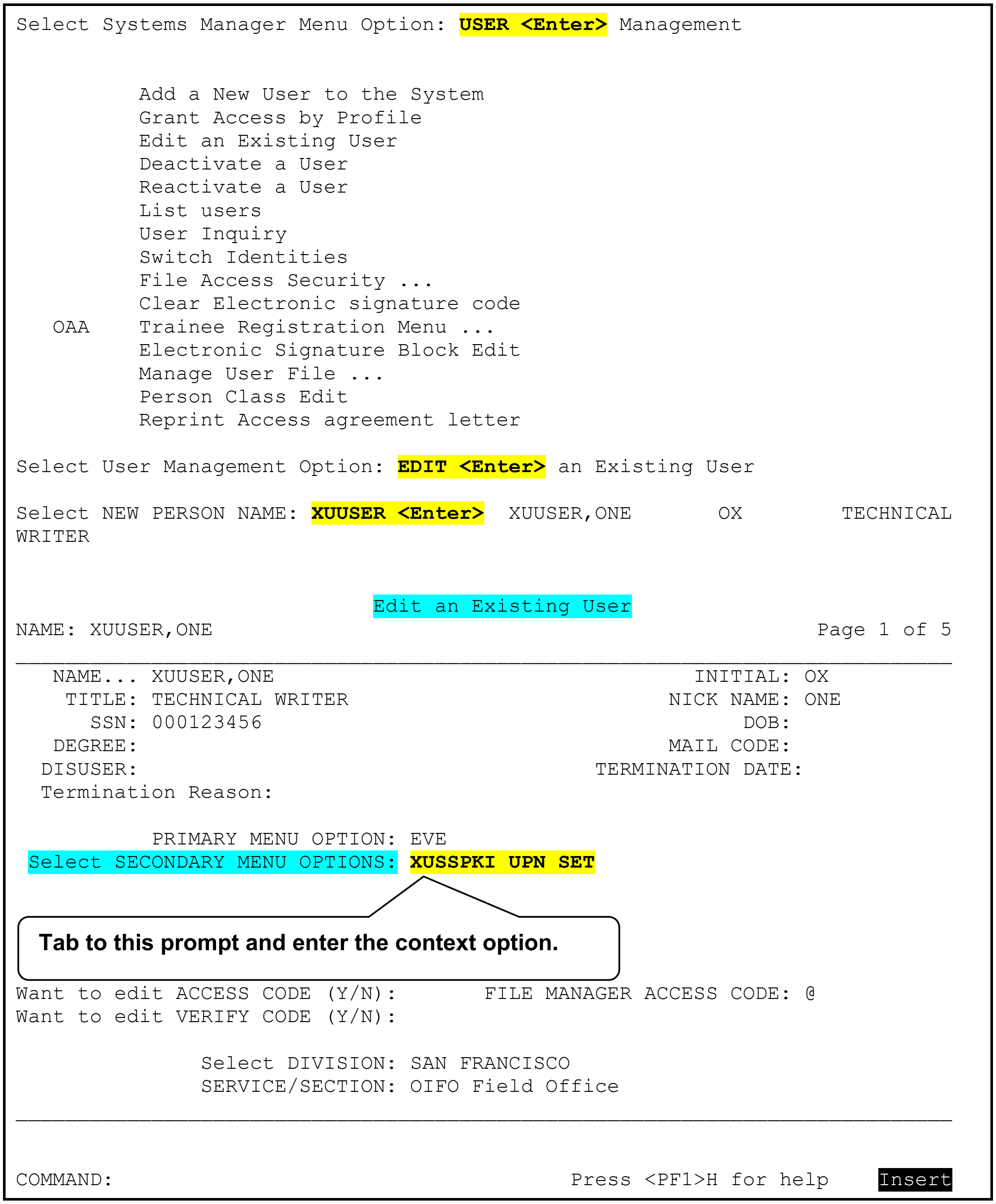
6.3.2.3 Assign the XUSSPKI UPN SET Option

The **ePCS Set SAN from PIV Card** [XUSSPKI UPN SET] option is the context option the RPC Broker uses for the DEA ePCS Utility when making remote procedure calls.

To assign the **ePCS Set SAN from PIV Card** [XUSSPKI UPN SET] option for each user, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **User Management** [XUSER] menu.
2. At the “Select User Management Option:” prompt, select the **Edit an Existing User** [XUSEREDIT] option.
3. At the “Select NEW PERSON NAME:” prompt, enter the user’s name.
4. In the “Edit an Existing User” main screen, tab down to the “Select SECONDARY MENU OPTIONS:” prompt, enter the **ePCS Set SAN from PIV Card** [XUSSPKI UPN SET] option.
5. (Optional) In the “SECONDARY MENU OPTIONS” popup screen, tab to “SYNONYM:” prompt and enter a synonym for this context option.
6. Tab to the “COMMAND:” prompt, enter **Close**. The “SECONDARY MENU OPTIONS” popup screen closes.
7. Tab to the “COMMAND:” prompt, enter **Exit**. The “Edit an Existing User” main screen closes.

Figure 59: DEA ePCS: Assigning the XUSSPKI UPN SET Option—Sample User Entries (1 of 2)

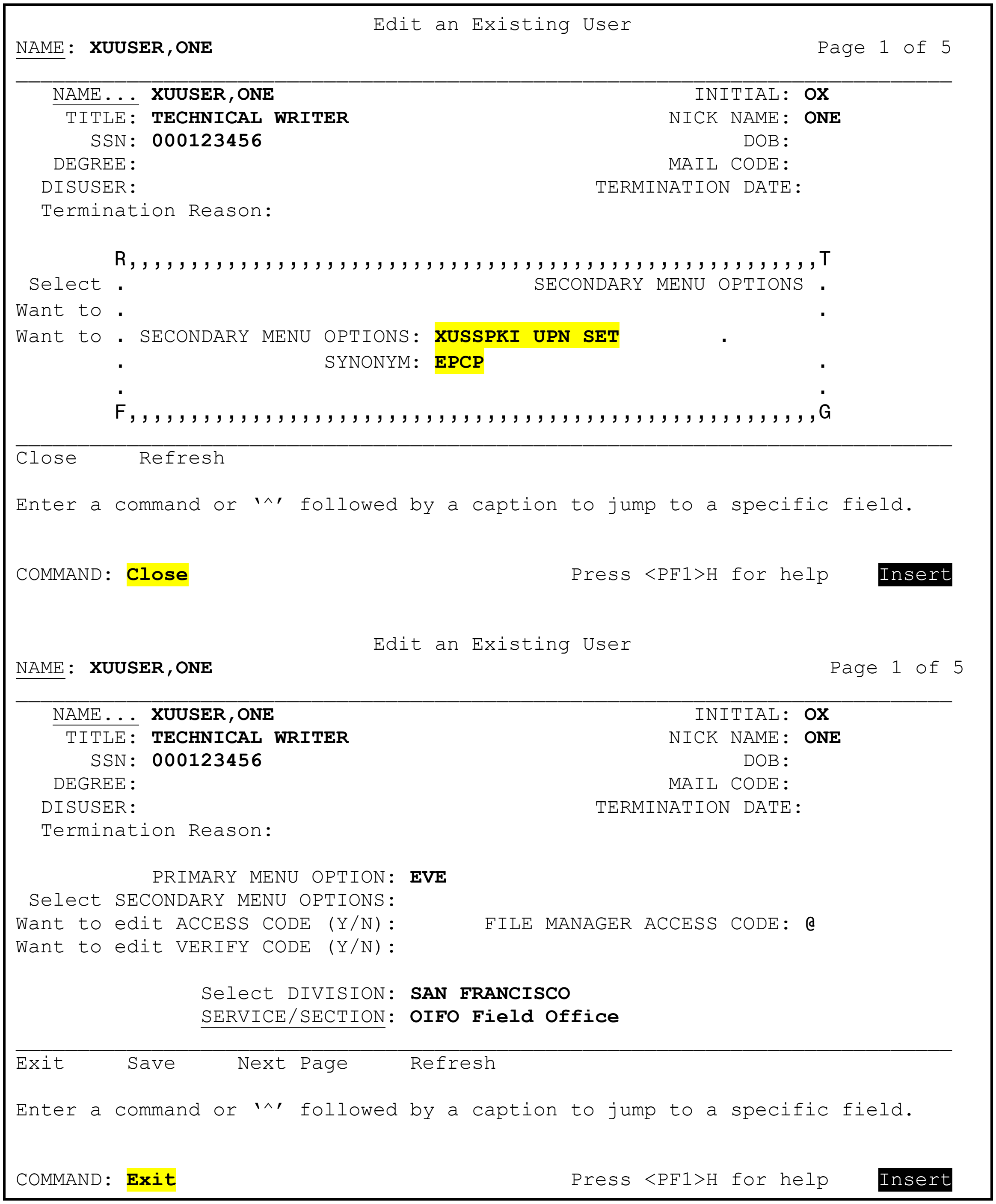


```
Select Systems Manager Menu Option: USER <Enter> Management


          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
   OAA    Trainee Registration Menu ...
          Electronic Signature Block Edit
          Manage User File ...
          Person Class Edit
          Reprint Access agreement letter

Select User Management Option: EDIT <Enter> an Existing User

Select NEW PERSON NAME: XUUSER <Enter>  XUUSER,ONE       OX        TECHNICAL
WRITER


                             Edit an Existing User
NAME: XUUSER,ONE                                                  Page 1 of 5
_____________________________________________________________________________
   NAME... XUUSER,ONE                                        INITIAL: OX
    TITLE: TECHNICAL WRITER                                NICK NAME: ONE
      SSN: 000123456                                             DOB:
   DEGREE:                                                 MAIL CODE:
  DISUSER:                                          TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS: XUSSPKI UPN SET
```

Tab to this prompt and enter the context option.

```
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office

_____________________________________________________________________________


COMMAND:                                      Press <PF1>H for help    Insert
```

Figure 60: DEA ePCS: Assigning the XUSSPKI UPN SET Option—Sample User Entries (2 of 2)



```
                            Edit an Existing User
NAME: XUUSER,ONE                                                    Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                       INITIAL: OX
    TITLE: TECHNICAL WRITER                               NICK NAME: ONE
      SSN: 000123456                                            DOB:
   DEGREE:                                                MAIL CODE:
  DISUSER:                                          TERMINATION DATE:
  Termination Reason:

        R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
 Select .                                 SECONDARY MENU OPTIONS .
Want to .                                                        .
Want to . SECONDARY MENU OPTIONS: XUSSPKI UPN SET        .
        .                SYNONYM: EPCP                           .
        .                                                        .
        F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
____________________________________________________________________________
Close     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: Close                                Press <PF1>H for help    Insert


                            Edit an Existing User
NAME: XUUSER,ONE                                                     Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                       INITIAL: OX
    TITLE: TECHNICAL WRITER                               NICK NAME: ONE
      SSN: 000123456                                            DOB:
   DEGREE:                                                MAIL CODE:
  DISUSER:                                          TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS:
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office
____________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: Exit                                 Press <PF1>H for help    Insert
```

6.4 Using the DEA ePCS Utility

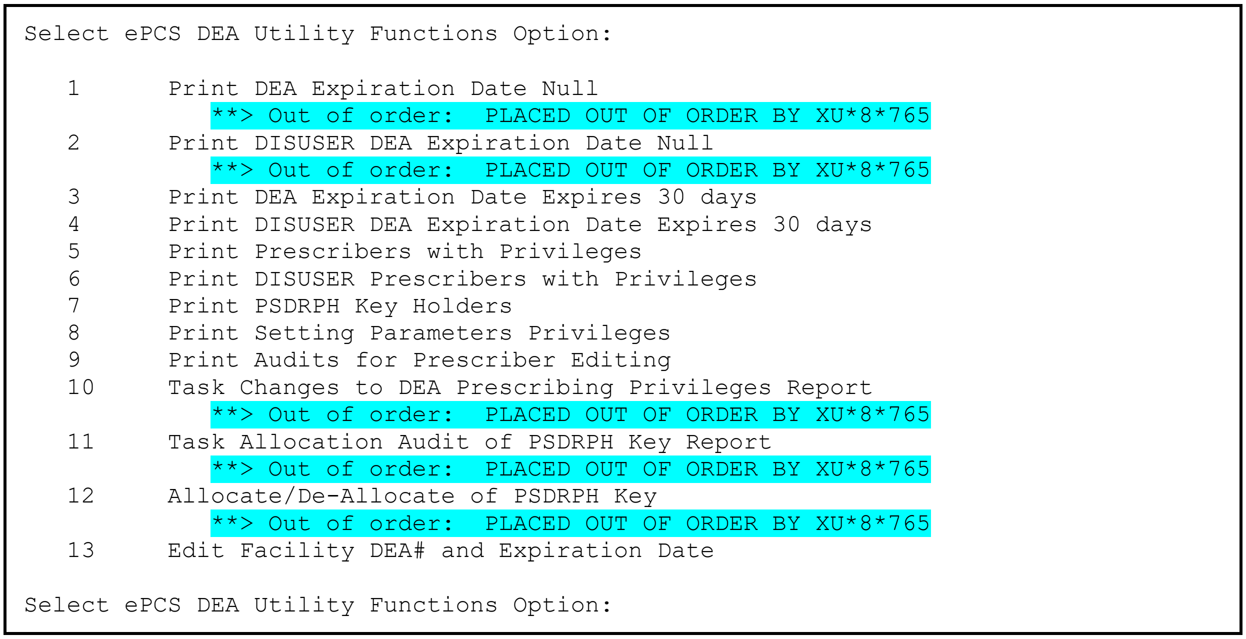
The DEA ePCS Utility consists of the following standalone menu and options, which are described in detail in the sections that follow:

- DEA ePCS Utility Functions Main Menu [XU EPCS UTILITY FUNCTIONS]
- Edit Facility DEA# and Expiration Date Option [XU EPCS EDIT DEA# AND XDATE]
- ePCS Edit Prescriber Data Option [XU EPCS EDIT DATA]
- ePCS Set SAN from PIV Card Option [XUSSPKI UPN SET]

6.4.1 DEA ePCS Utility Functions Main Menu

Released with Kernel patch XU*8.0*580, the **ePCS DEA Utility Functions** main menu [XU EPCS UTILITY FUNCTIONS] main menu is a standalone menu that is *not* linked to any other Kernel menus. It includes the options shown in Figure 61, which are described in Table 10:

Figure 61: DEA ePCS: DEA ePCS Utility Functions Main Menu [XU EPCS UTILITY FUNCTIONS]



```
Select ePCS DEA Utility Functions Option:

   1      Print DEA Expiration Date Null
             **> Out of order:  PLACED OUT OF ORDER BY XU*8*765
   2      Print DISUSER DEA Expiration Date Null
             **> Out of order:  PLACED OUT OF ORDER BY XU*8*765
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
             **> Out of order:  PLACED OUT OF ORDER BY XU*8*765
   11     Task Allocation Audit of PSDRPH Key Report
             **> Out of order:  PLACED OUT OF ORDER BY XU*8*765
   12     Allocate/De-Allocate of PSDRPH Key
             **> Out of order:  PLACED OUT OF ORDER BY XU*8*765
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option:
```

NOTE: The following options were placed out of order with Kernel Patch XU*8.0*765 and replaced by functionality added to the **ePCS DEA Utility Functions** [PSO EPCS UTILITY FUNCTIONS] menu:

- `1       Print DEA Expiration Date Null`
- `2       Print DISUSER DEA Expiration Date Null`
- `10      Task Changes to DEA Prescribing Privileges Report`
- `11      Task Allocation Audit of PSDRPH Key Report`
- `12      Allocate/De-Allocate of PSDRPH Key`

Table 10: DEA ePCS Utility—Main Menu Options

Option Name	Option Menu Text	Description
XU EPCS UTILITY FUNCTIONS	ePCS DEA Utility Functions	This is the main menu for the DEA ePCS Utility. It includes the following options: • **XU EPCS EXP DATE** • **XU EPCS DISUSER EXP DATE** • **XU EPCS XDATE EXPIRES** • **XU EPCS DISUSER XDATE EXPIRES** • **XU EPCS PRIVS** • **XU EPCS DISUSER PRIVS** • **XU EPCS PSDRPH** • **XU EPCS SET PARMS** • **XU EPCS PRINT EDIT AUDIT** • **XU EPCS LOGICAL ACCESS** • **XU EPCS PSDRPH AUDIT** • **XU EPCS PSDRPH KEY** • **XU EPCS EDIT DEA# AND XDATE**
XU EPCS EXP DATE (See Section 6.4.2.)	Print DEA Expiration Date Null	This option prints all active users with an unpopulated DEA# and DEA EXPIRATION DATE. This option prints the following data: • NAME • DEA# • DEA EXPIRATION DATE
XU EPCS DISUSER EXP DATE	Print DISUSER DEA Expiration Date Null	This option prints all DISUSERed users with an unpopulated DEA# and

Option Name	Option Menu Text	Description
(See Section 6.4.3.)		DEA EXPIRATION DATE. This option prints the following data: • NAME • DEA# • TERMINATION DATE • DEA EXPIRATION DATE
XU EPCS XDATE EXPIRES (See Section 6.4.4.)	Print DEA Expiration Date Expires 30 days	This option prints all active users with DEA # and where the DEA EXPIRATION DATE expires within 30 days. This option prints the following data: • NAME • DEA# • DEA EXPIRATION DATE
XU EPCS DISUSER XDATE EXPIRES (See Section 6.4.5.)	Print DISUSER DEA Expiration Date Expires 30 days	This option prints all DISUSERed users with DEA # and where the DEA EXPIRATION DATE expires within 30 days. This option prints the following data: • NAME • DEA# • DEA EXPIRATION DATE
XU EPCS PRIVS (See Section 6.4.6.)	Print Prescribers with Privileges	This option prints all active users who have privileges to any of the SCHEDULEs II through V and who have a DEA# or VA#. This option prints the following data: • NAME • **DUZ** • DEA# • VA# • SCHEDULESs
XU EPCS DISUSER PRIVS (See Section 6.4.7.)	Print DISUSER Prescribers with Privileges	This option prints all DISUSERed users who have privileges to any of the SCHEDULEs II through V and who have a DEA# or VA#. This option prints the following data: • NAME • **DUZ** • DEA#

Option Name	Option Menu Text	Description
		• TERMINATION DATE • VA# • SCHEDULESs
XU EPCS PSDRPH (See Section 6.4.8.)	Print PSDRPH Key Holders	This option prints all active users holding the PSDRPH security key. This report sorts by Division, and within DIVISION, it sorts by NAME. This option prints the following data: • NAME • **DUZ** • GIVEN BY (Person Who Assigned Key) • DATE GIVEN (Date Assigned)
XU EPCS SET PARMS (See Section 6.4.9.)	Print Setting Parameters Privileges	This option prints all active users holding the XUEPCSEDIT security key. This option identifies individuals responsible for setting the parameters.
XU EPCS PRINT EDIT AUDIT (See Section 6.4.10.)	Print Audits for Prescriber Editing	This option prints information related to the editing of prescriber information.
XU EPCS LOGICAL ACCESS (See Section 6.4.11.)	Task Changes to DEA Prescribing Privileges Report	This tasked option prints the setting or change to DEA prescribing privileges related to issuance of a controlled substance prescription. This option only prints data from the previous day and with data that has been modified. The data is retrieved from the XUEPCS DATA (#8991.6) file. This option should be scheduled to run on a daily basis.
XU EPCS PSDRPH AUDIT (See Section 6.4.12.)	Task Allocation Audit of PSDRPH Key Report	This tasked option prints the allocation of the PSDRPH security key. This option only prints data from the previous day and with data that has been modified. The report prints data for the archive XUEPCS PSDRPH AUDIT (#8991.7) file. This option should be scheduled to run on a daily basis.
XU EPCS PSDRPH KEY (See Section 6.4.13.)	Allocate/De-Allocate of PSDRPH Key	This option allocates or de-allocates the PSDRPH security key.

Option Name	Option Menu Text	Description
XU EPCS EDIT DEA# AND XDATE (See Section 6.4.14.)	Edit Facility DEA# and Expiration Date	This option edits the FACILITY DEA NUMBER (#52) and FACILITY DEA EXPIRATION DATE (#52.1) fields in the INSTITUTION (#4) file.

6.4.2 Print DEA Expiration Date Null Option

The **Print DEA Expiration Date Null** [XU EPCS EXP DATE] option prints all active users from the NEW PERSON (#200) file with the following field values:

- DEA# (#53.2) field—**NULL** (*unpopulated*).
- DEA EXPIRATION DATE (#747.44)—Not **NULL** (*populated*).

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- DEA# (#53.2)
- DEA EXPIRATION DATE (#747.44)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 62: DEA ePCS: Print DEA Expiration Date Null Option—Sample User Entries and Report

```
Select Systems Manager Menu Option: EPCS <Enter> ePCS DEA Utility Functions

   1       Print DEA Expiration Date Null
   2       Print DISUSER DEA Expiration Date Null
   3       Print DEA Expiration Date Expires 30 days
   4       Print DISUSER DEA Expiration Date Expires 30 days
   5       Print Prescribers with Privileges
   6       Print DISUSER Prescribers with Privileges
   7       Print PSDRPH Key Holders
   8       Print Setting Parameters Privileges
   9       Print Audits for Prescriber Editing
   10      Task Changes to DEA Prescribing Privileges Report
   11      Task Allocation Audit of PSDRPH Key Report
   12      Allocate/De-Allocate of PSDRPH Key
   13      Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 1 <Enter> Print DEA Expiration Date Null
START WITH NAME: FIRST// <Enter>
DEVICE: <Enter>  HOME  (CRT)    Right Margin: 80// <Enter>
NULL 'DEA EXPIRATION DATE'                     APR 15,2013  16:53    PAGE 1
                                                    DEA
                                                    EXPIRATION
NAME                                   DEA#         DATE
--------------------------------------------------------------------------------


XUUSER,EIGHT                           AK1662673
XUUSER,ELEVEN                          MT0300777
XUUSER,FIVE                            BH2942628
XUUSER,FOUR                            AK2984082
XUUSER,FOURTEEN                        AG5333745
XUUSER,NINE                            BB1770773
XUUSER,ONE                             SF0963226
XUUSER,SEVEN                           AP8348458
XUUSER,SIX                             AM7446001
XUUSER,TEN                             BD9270911
XUUSER,THIRTEEN                        FC2158548
XUUSER,THREE                           FS2138572
XUUSER,TWELVE                          AR3287946
XUUSER,TWO                             BG4740850
.
.
.
```

6.4.3 Print DISUSER DEA Expiration Date Null Option

The **Print DISUSER DEA Expiration Date Null** [XU EPCS DISUSER EXP DATE] option prints all DISUSERed users from the NEW PERSON (#200) file with the following field values:

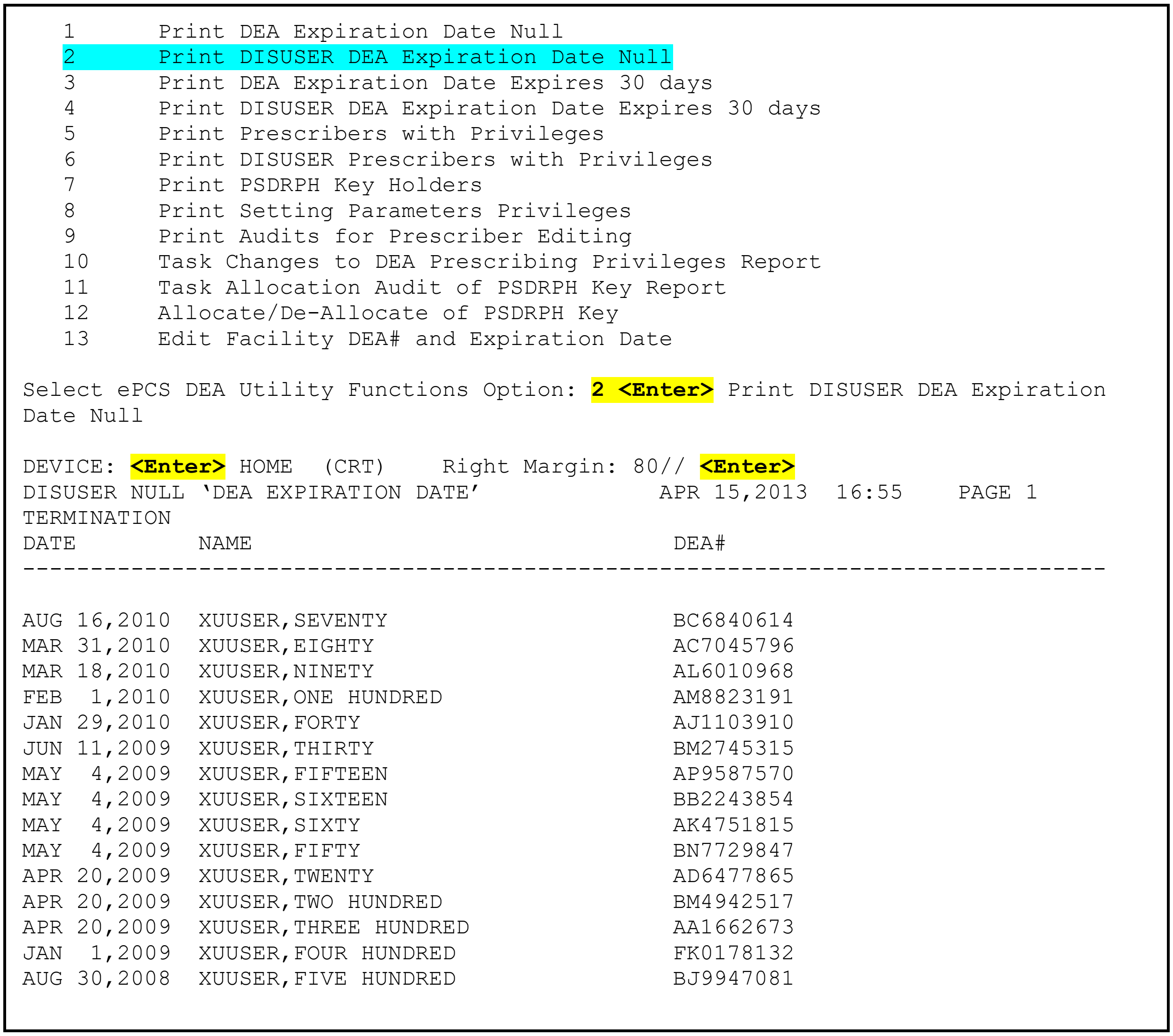
- DEA# (#53.2)— **NULL** (*unpopulated*).
- DEA EXPIRATION DATE (#747.44)—Not **NULL** (*populated*).

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- DEA# (#53.2)
- TERMINATION DATE (#9.2)
- DEA EXPIRATION DATE (#747.44)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 63: DEA ePCS: Print DISUSER DEA Expiration Date Null Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 2 <Enter> Print DISUSER DEA Expiration
Date Null

DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
DISUSER NULL 'DEA EXPIRATION DATE'             APR 15,2013  16:55    PAGE 1
TERMINATION
DATE         NAME                                  DEA#
--------------------------------------------------------------------------------


AUG 16,2010  XUUSER,SEVENTY                        BC6840614
MAR 31,2010  XUUSER,EIGHTY                         AC7045796
MAR 18,2010  XUUSER,NINETY                         AL6010968
FEB  1,2010  XUUSER,ONE HUNDRED                    AM8823191
JAN 29,2010  XUUSER,FORTY                          AJ1103910
JUN 11,2009  XUUSER,THIRTY                         BM2745315
MAY  4,2009  XUUSER,FIFTEEN                        AP9587570
MAY  4,2009  XUUSER,SIXTEEN                        BB2243854
MAY  4,2009  XUUSER,SIXTY                          AK4751815
MAY  4,2009  XUUSER,FIFTY                          BN7729847
APR 20,2009  XUUSER,TWENTY                         AD6477865
APR 20,2009  XUUSER,TWO HUNDRED                    BM4942517
APR 20,2009  XUUSER,THREE HUNDRED                  AA1662673
JAN  1,2009  XUUSER,FOUR HUNDRED                   FK0178132
AUG 30,2008  XUUSER,FIVE HUNDRED                   BJ9947081
```

6.4.4 Print DEA Expiration Date Expires 30 days Option

The **Print DEA Expiration Date Expires 30 days**[XU EPCS XDATE EXPIRES] option prints all active users from the NEW PERSON (#200) file with the following field values:

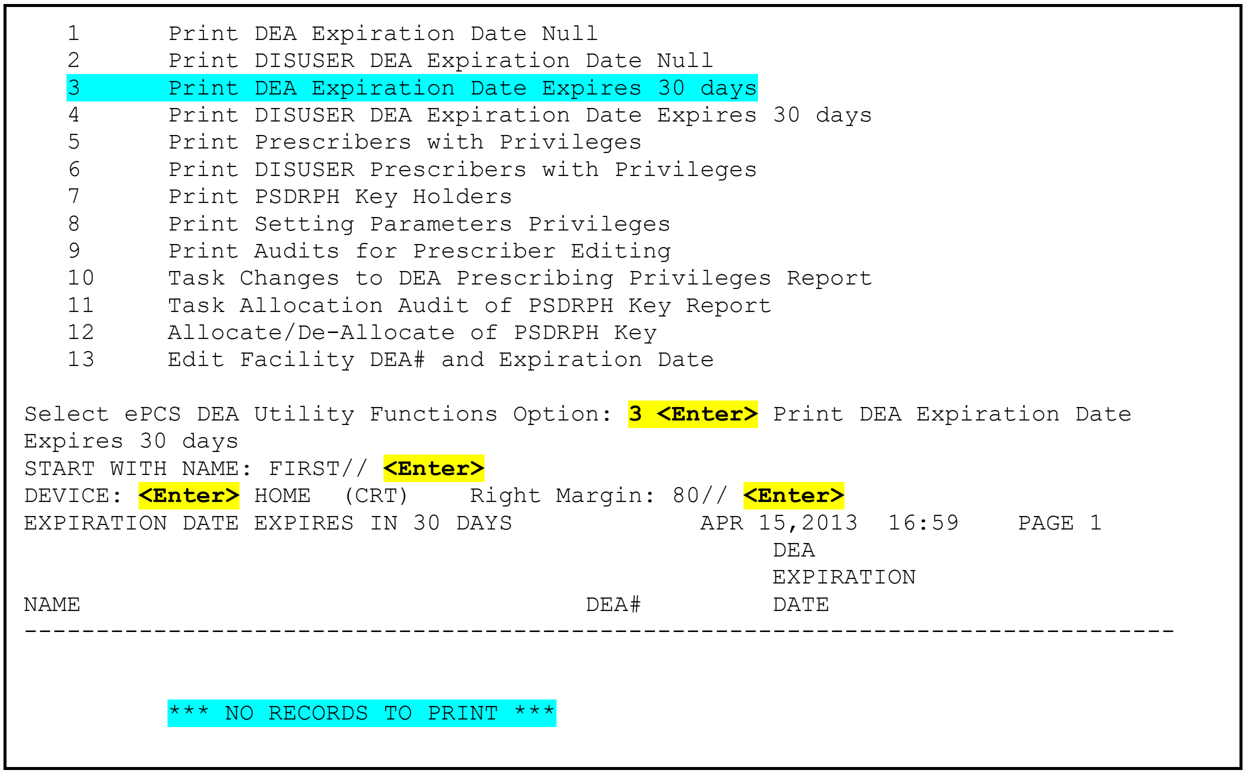
- DEA# (#53.2) field—Not **NULL** (*populated*).
- DEA EXPIRATION DATE (#747.44) field—Date expires within **30** days.

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- DEA# (#53.2)
- DEA EXPIRATION DATE (#747.44)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 64: DEA ePCS: Print DEA Expiration Date Expires 30 days Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 3 <Enter> Print DEA Expiration Date
Expires 30 days
START WITH NAME: FIRST// <Enter>
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
EXPIRATION DATE EXPIRES IN 30 DAYS            APR 15,2013  16:59    PAGE 1
                                                   DEA
                                                   EXPIRATION
NAME                                   DEA#         DATE
--------------------------------------------------------------------------------


          *** NO RECORDS TO PRINT ***
```

6.4.5 Print DISUSER DEA Expiration Date Expires 30 days Option

The **Print DISUSER DEA Expiration Date Expires 30 days** [XU EPCS DISUSER XDATE EXPIRES] option prints all DISUSERed users from the NEW PERSON (#200) file with the following field values:

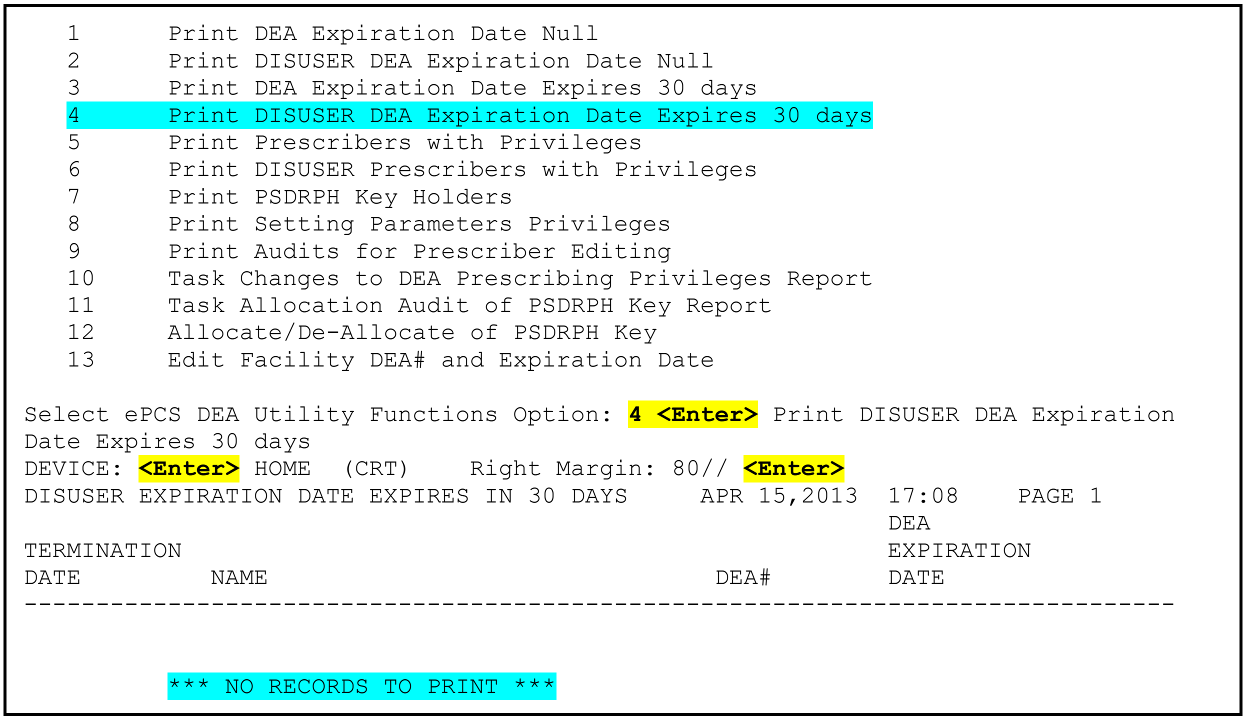
- DEA# (#53.2) field—Not **NULL** (*populated*).
- DEA EXPIRATION DATE (#747.44) field—Date expires within **30** days.

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- DEA# (#53.2)
- DEA EXPIRATION DATE (#747.44)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 65: DEA ePCS: Print DISUSER DEA Expiration Date Expires 30 days Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 4 <Enter> Print DISUSER DEA Expiration
Date Expires 30 days
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
DISUSER EXPIRATION DATE EXPIRES IN 30 DAYS     APR 15,2013  17:08    PAGE 1
                                                            DEA
TERMINATION                                                 EXPIRATION
DATE         NAME                                  DEA#        DATE
--------------------------------------------------------------------------------


          *** NO RECORDS TO PRINT ***
```

6.4.6 Print Prescribers with Privileges Option

The **Print Prescribers with Privileges** [XU EPCS PRIVS] option prints all active users from the NEW PERSON (#200) file who have privileges to any of the SCHEDULEs **II** through **V** and who have a DEA# or VA#.

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- **DUZ**—Internal Entry Number (IEN) for the user in the NEW PERSON (#200) file
- DEA# (#53.2)
- VA# (#53.3)

- SCHEDULEs:
  - SCHEDULE II NARCOTIC (#55.1)
  - SCHEDULE II NON-NARCOTIC (#55.2)
  - SCHEDULE III NARCOTIC (#55.3)
  - SCHEDULE III NON-NARCOTIC (#55.4)
  - SCHEDULE IV (#55.5)
  - SCHEDULE V (#55.6)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 66: DEA ePCS: Print Prescribers with Privileges Option—Sample User Entries and Report

```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 5 <Enter> Print Prescribers with
Privileges
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
PRESCRIBERS WITH PRIVILEGES                    APR 15,2013  17:13    PAGE 1
NAME                            DUZ           DEA#          VA#
--------------------------------------------------------------------------------


        DIVISION: ALBANY, NY VAMC
XUUSER,ONE                      520736424     AA1234563
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
         SCHEDULE III NON:   Yes
         SCHEDULE IV:        Yes
         SCHEDULE V:
        DIVISION: CHEYENNE VAMC
XUUSER,TWO                      520629114                   AV4538419
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:
.
.
.
```

6.4.7 Print DISUSER Prescribers with Privileges Option

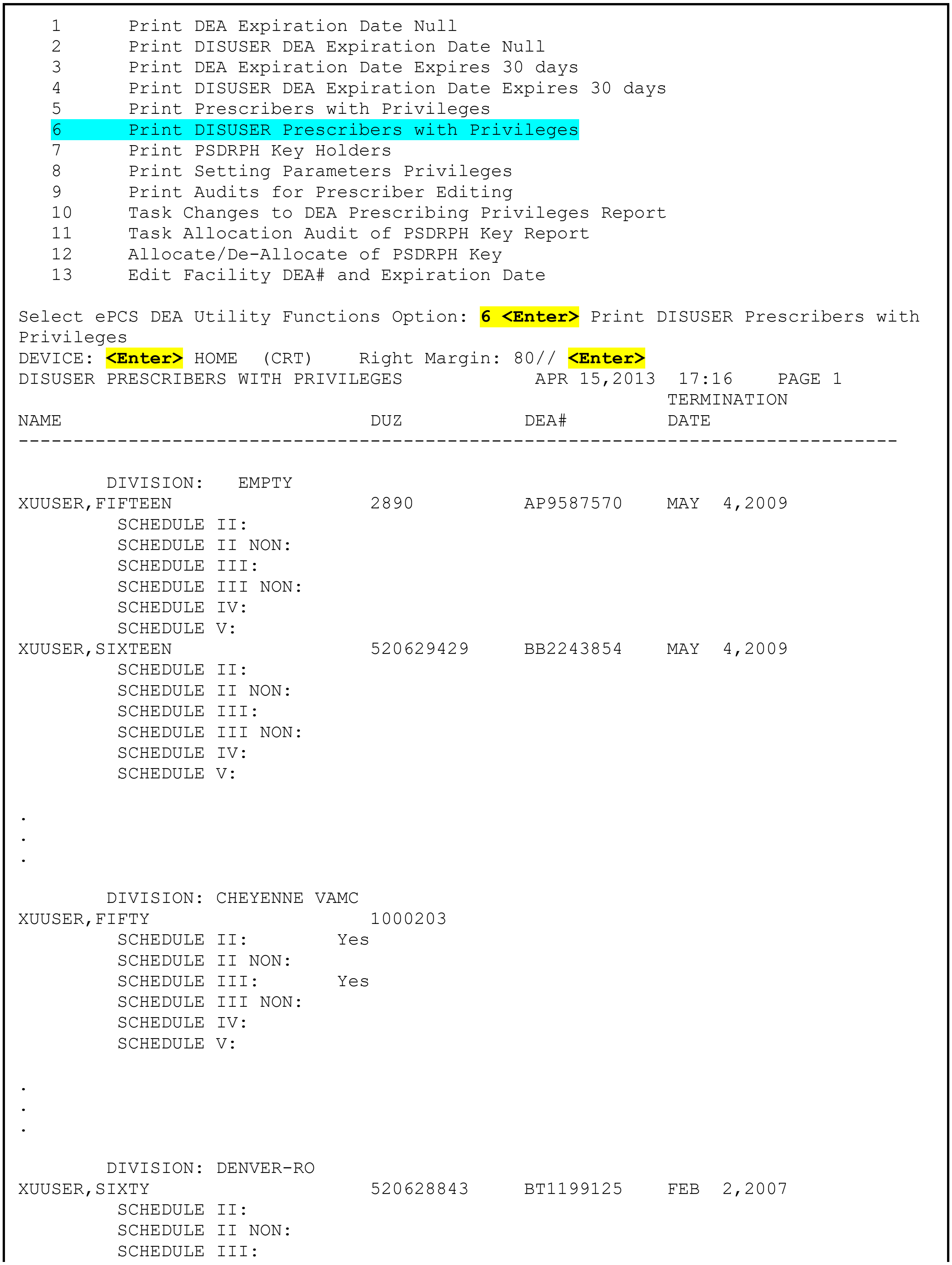
The **Print DISUSER Prescribers with Privileges** [XU EPCS DISUSER PRIVS] option prints all DISUSERed users who have privileges to any of the SCHEDULEs II through V and who have a DEA# or VA#.

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- **DUZ**—Internal Entry Number (IEN) for the user in the NEW PERSON (#200) file
- DEA# (#53.2)
- TERMINATION DATE (#9.2)
- VA# (#53.3) (DIVISION)
- SCHEDULEs:
    - SCHEDULE II NARCOTIC (#55.1)
    - SCHEDULE II NON-NARCOTIC (#55.2)
    - SCHEDULE III NARCOTIC (#55.3)
    - SCHEDULE III NON-NARCOTIC (#55.4)
    - SCHEDULE IV (#55.5)
    - SCHEDULE V (#55.6)

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 67: DEA ePCS: Print DISUSER Prescribers with Privileges Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 6 <Enter> Print DISUSER Prescribers with
Privileges
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
DISUSER PRESCRIBERS WITH PRIVILEGES            APR 15,2013  17:16    PAGE 1
                                                           TERMINATION
NAME                            DUZ           DEA#         DATE
--------------------------------------------------------------------------------


        DIVISION:   EMPTY
XUUSER,FIFTEEN                  2890          AP9587570    MAY  4,2009
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:
XUUSER,SIXTEEN                  520629429     BB2243854    MAY  4,2009
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:

.
.
.

        DIVISION: CHEYENNE VAMC
XUUSER,FIFTY                    1000203
         SCHEDULE II:        Yes
         SCHEDULE II NON:
         SCHEDULE III:       Yes
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:

.
.
.

        DIVISION: DENVER-RO
XUUSER,SIXTY                    520628843     BT1199125    FEB  2,2007
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
```

```
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:
XUUSER,SEVENTY                  520628775     AH9494852    FEB 12,1999
         SCHEDULE II:
         SCHEDULE II NON:
         SCHEDULE III:
         SCHEDULE III NON:
         SCHEDULE IV:
         SCHEDULE V:
XUUSER,EIGHTY                   520628129     BA4578893    OCT 12,1990
         SCHEDULE II:        Yes
         SCHEDULE II NON:    Yes
         SCHEDULE III:       Yes
         SCHEDULE III NON:   Yes
         SCHEDULE IV:        Yes
         SCHEDULE V:         Yes
.
.
.
```

6.4.8 Print PSDRPH Key Holders Option

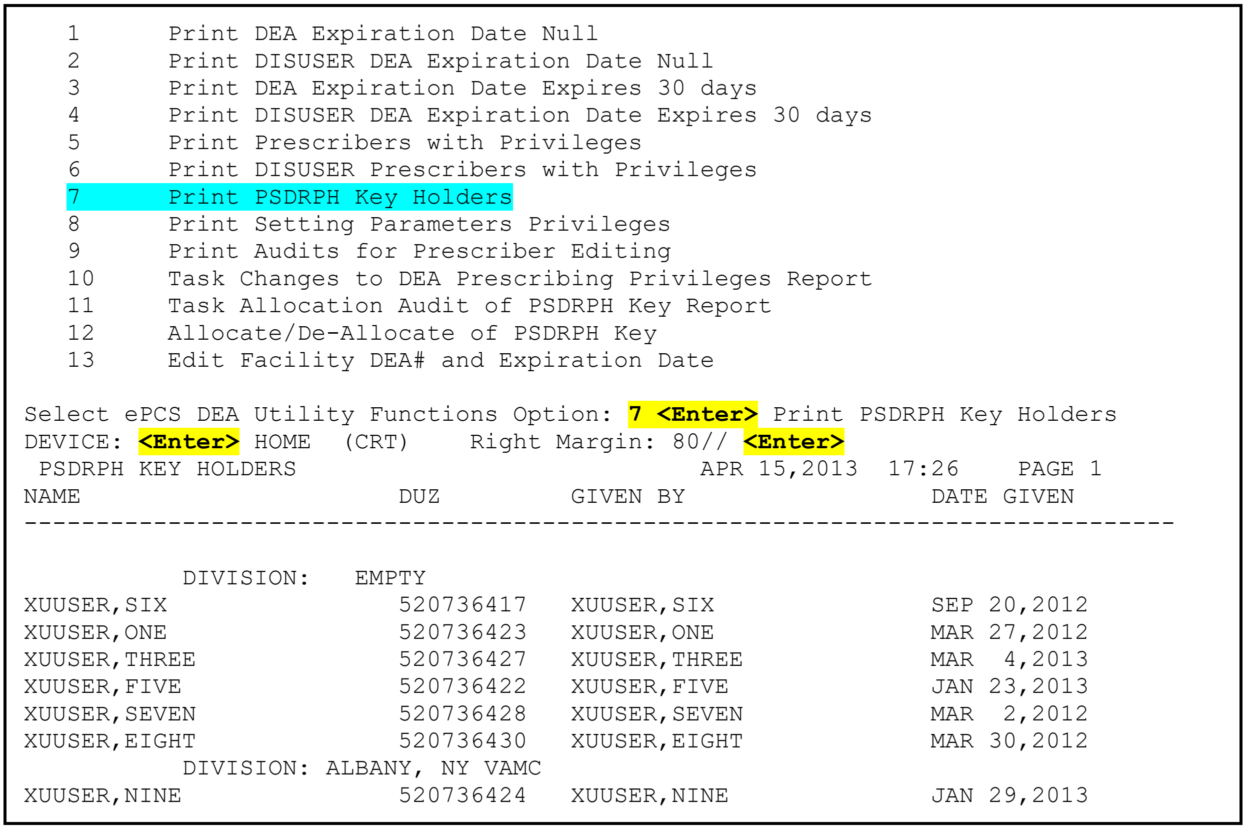
The **Print PSDRPH Key Holders** [XU EPCS PSDRPH] option prints all active users holding the PSDRPH security key. This report sorts by Division, and within Division, it sorts by Name.

This option prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- **DUZ**—Internal Entry Number (IEN) for the user in the NEW PERSON (#200) file
- GIVEN BY (#1) subfield of the KEYS (#51) Multiple: Person who assigned the PSDRPH security key
- DATE GIVEN (#2) subfield of the KEYS (#51) Multiple: Date assigned

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 68: DEA ePCS: Print PSDRPH Key Holders Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 7 <Enter> Print PSDRPH Key Holders
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
 PSDRPH KEY HOLDERS                          APR 15,2013  17:26    PAGE 1
NAME                      DUZ         GIVEN BY                 DATE GIVEN
--------------------------------------------------------------------------------


           DIVISION:   EMPTY
XUUSER,SIX                520736417   XUUSER,SIX               SEP 20,2012
XUUSER,ONE                520736423   XUUSER,ONE               MAR 27,2012
XUUSER,THREE              520736427   XUUSER,THREE             MAR  4,2013
XUUSER,FIVE               520736422   XUUSER,FIVE              JAN 23,2013
XUUSER,SEVEN              520736428   XUUSER,SEVEN             MAR  2,2012
XUUSER,EIGHT              520736430   XUUSER,EIGHT             MAR 30,2012
           DIVISION: ALBANY, NY VAMC
XUUSER,NINE               520736424   XUUSER,NINE              JAN 29,2013
```

6.4.9 Print Setting Parameters Privileges Option

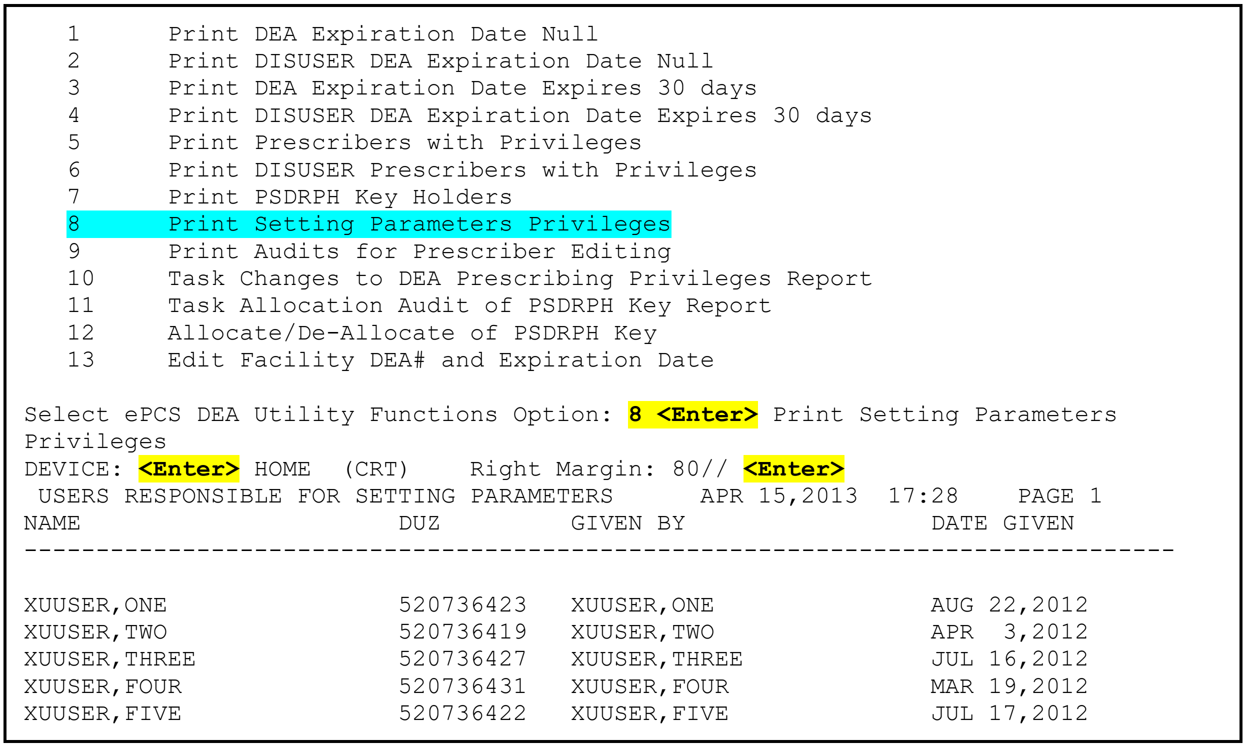
The **Print Setting Parameters Privileges** [XU EPCS SET PARMS] option prints all active users holding the XUEPCSEDIT security key.

This option identifies individuals responsible for setting the parameters. It prints the following data from the NEW PERSON (#200) file:

- NAME (#.01)
- **DUZ**—Internal Entry Number (IEN) for the user in the NEW PERSON (#200) file
- GIVEN BY (#1) subfield of the KEYS (#51) Multiple: Person who assigned the PSDRPH security key
- DATE GIVEN (#2) subfield of the KEYS (#51) Multiple: Date assigned

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 69: DEA ePCS: Print Setting Parameters Privileges Option—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 8 <Enter> Print Setting Parameters
Privileges
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>
 USERS RESPONSIBLE FOR SETTING PARAMETERS      APR 15,2013  17:28    PAGE 1
NAME                      DUZ         GIVEN BY                 DATE GIVEN
--------------------------------------------------------------------------------


XUUSER,ONE                520736423   XUUSER,ONE               AUG 22,2012
XUUSER,TWO                520736419   XUUSER,TWO               APR  3,2012
XUUSER,THREE              520736427   XUUSER,THREE             JUL 16,2012
XUUSER,FOUR               520736431   XUUSER,FOUR              MAR 19,2012
XUUSER,FIVE               520736422   XUUSER,FIVE              JUL 17,2012
```

6.4.10 Print Audits for Prescriber Editing Option

The **Print Audits for Prescriber Editing** [XU EPCS PRINT EDIT AUDIT] option prints information related to the editing of prescriber information.

The data for this report is retrieved from the XUEPCS DATA (#8991.6) file. It prints the following data:

- DATE/TIME EDITED (#.06)
- NAME (#.01)—This is the name of user edited.
- EDITED BY (#.02)—This is the name of user who edited the data.
- FIELD EDITED (#.03)
- ORIGINAL DATA (#.04)
- EDITED DATA (#.05)

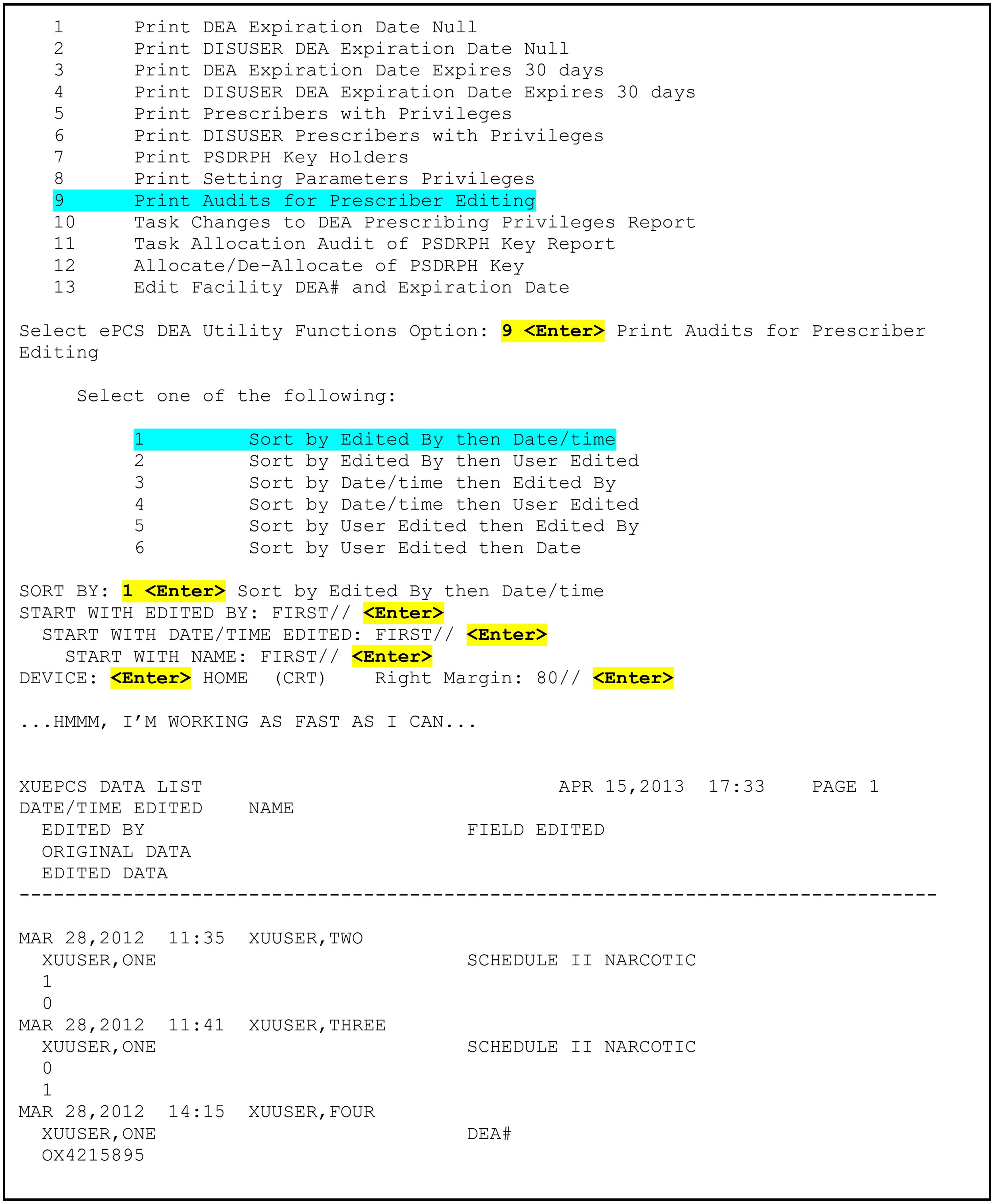
You can sort the data by any of the following data:

- Edited By then Date/Time
- Edited By then User Edited
- Date/Time then Edited By
- Date/Time then User Edited

- User Edited then Edited By
- User Edited then Date

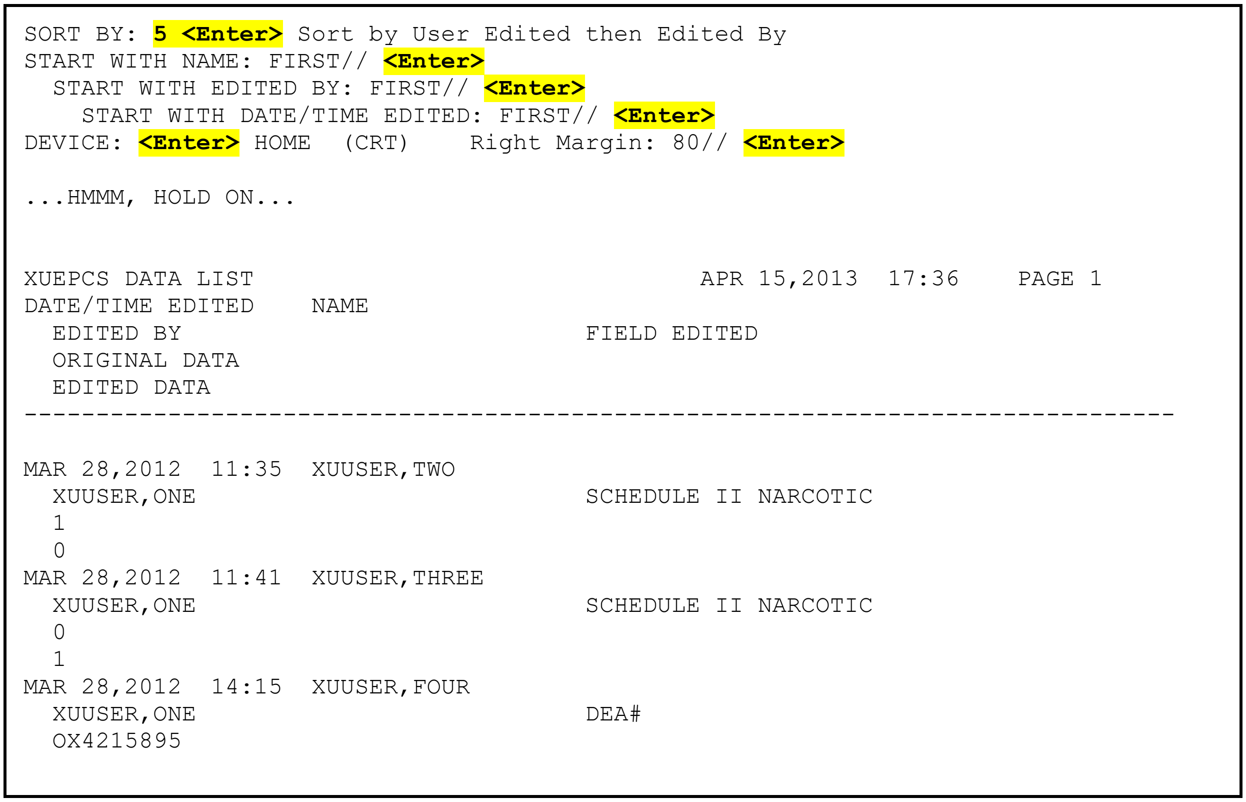
NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 70: DEA ePCS: Print Audits for Prescriber Editing Option: Sort by *Edited By then Date/time*—Sample User Entries and Report



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 9 <Enter> Print Audits for Prescriber
Editing

     Select one of the following:

          1         Sort by Edited By then Date/time
          2         Sort by Edited By then User Edited
          3         Sort by Date/time then Edited By
          4         Sort by Date/time then User Edited
          5         Sort by User Edited then Edited By
          6         Sort by User Edited then Date

SORT BY: 1 <Enter> Sort by Edited By then Date/time
START WITH EDITED BY: FIRST// <Enter>
  START WITH DATE/TIME EDITED: FIRST// <Enter>
    START WITH NAME: FIRST// <Enter>
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>

...HMMM, I'M WORKING AS FAST AS I CAN...


XUEPCS DATA LIST                               APR 15,2013  17:33    PAGE 1
DATE/TIME EDITED    NAME
  EDITED BY                            FIELD EDITED
  ORIGINAL DATA
  EDITED DATA
--------------------------------------------------------------------------------

MAR 28,2012  11:35  XUUSER,TWO
  XUUSER,ONE                           SCHEDULE II NARCOTIC
  1
  0
MAR 28,2012  11:41  XUUSER,THREE
  XUUSER,ONE                           SCHEDULE II NARCOTIC
  0
  1
MAR 28,2012  14:15  XUUSER,FOUR
  XUUSER,ONE                           DEA#
  OX4215895
```

Figure 71: DEA ePCS: Print Audits for Prescriber Editing Option: Sort by *User Edited then Edited By*—Sample User Entries and Report



```
SORT BY: 5 <Enter> Sort by User Edited then Edited By
START WITH NAME: FIRST// <Enter>
  START WITH EDITED BY: FIRST// <Enter>
    START WITH DATE/TIME EDITED: FIRST// <Enter>
DEVICE: <Enter> HOME  (CRT)    Right Margin: 80// <Enter>

...HMMM, HOLD ON...


XUEPCS DATA LIST                               APR 15,2013  17:36    PAGE 1
DATE/TIME EDITED    NAME
  EDITED BY                           FIELD EDITED
  ORIGINAL DATA
  EDITED DATA
--------------------------------------------------------------------------------

MAR 28,2012  11:35  XUUSER,TWO
  XUUSER,ONE                          SCHEDULE II NARCOTIC
  1
  0
MAR 28,2012  11:41  XUUSER,THREE
  XUUSER,ONE                          SCHEDULE II NARCOTIC
  0
  1
MAR 28,2012  14:15  XUUSER,FOUR
  XUUSER,ONE                          DEA#
  OX4215895
```

6.4.11 Task Changes to DEA Prescribing Privileges Report Option

CAUTION: Verify that the XUEPCS REPORT DEVICE parameter has been set before using this option.

To set the parameter, see the "Set the XUEPCS REPORT DEVICE Parameter" section.

The **Task Changes to DEA Prescribing Privileges Report** [XU EPCS LOGICAL ACCESS] option prints the setting or change to DEA prescribing privileges related to issuance of a controlled substance prescription.

The option only prints data from the previous day and with data that has been modified. The data is retrieved from the XUEPCS DATA (#8991.6) file.

This option should be scheduled to run on a daily basis via TaskMan. The option only prints data from the *previous* day and with *data that has been modified*. The data is retrieved from the XUEPCS DATA (#8991.6) file.

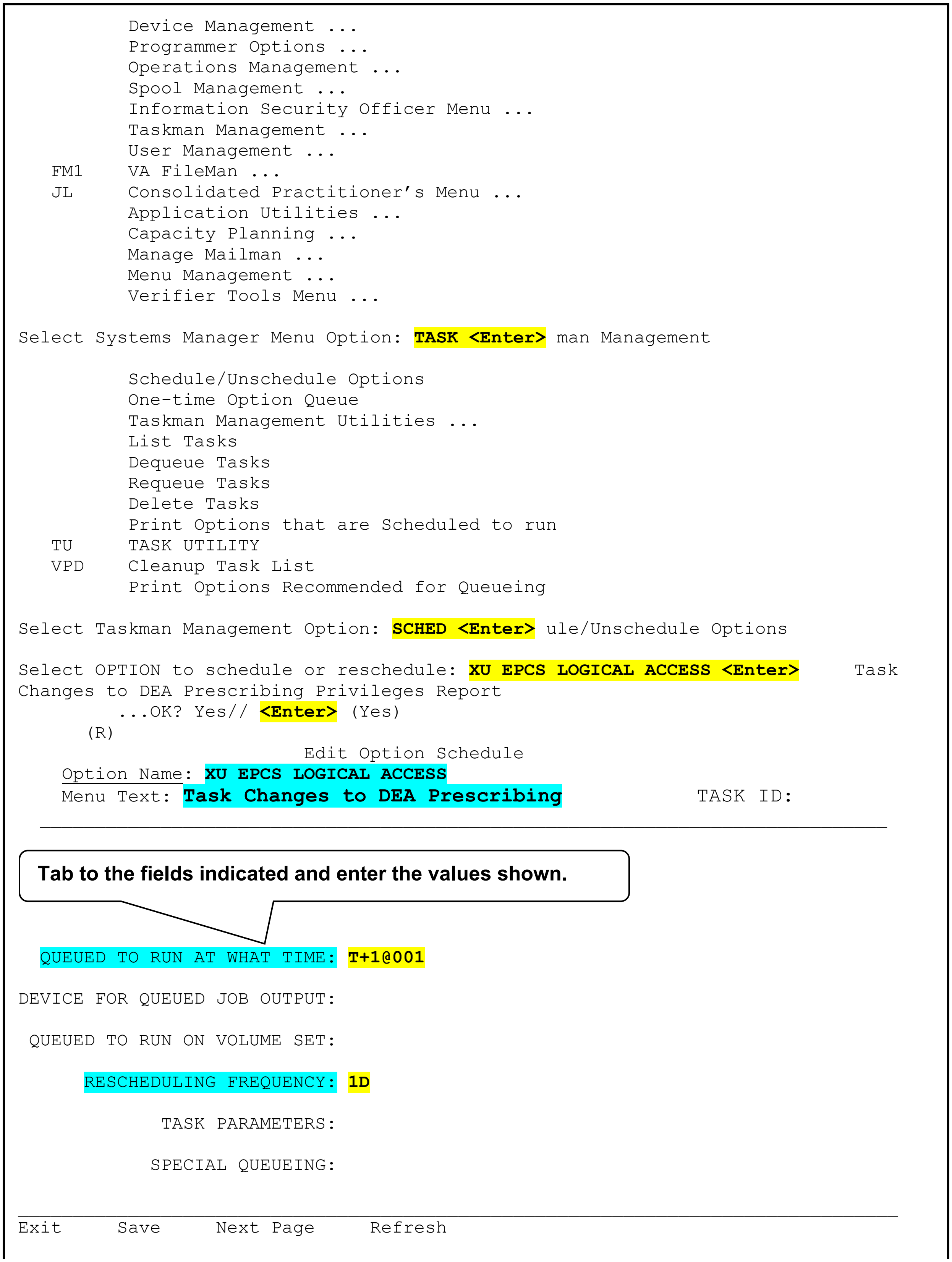
NOTE: No data is displayed to the screen; the data is printed to the device indicated by the XUEPCS REPORT DEVICE parameter.

NOTE: This option was released with Kernel patch XU*8.0*580.

To schedule the option to run daily using TaskMan, perform the following procedure:

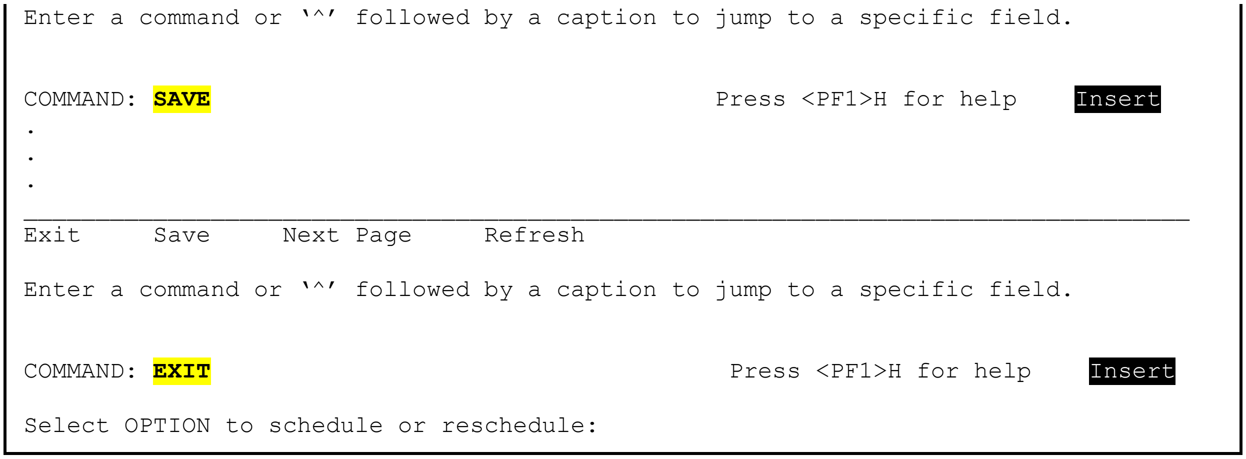
1. From the **Systems Manager Menu** [EVE], select the **Taskman Management** [XUTM MGR] option.
2. At the “Select Taskman Management Option:” prompt, select the **Schedule/Unschedule Options** [XUTM SCHEDULE] option.
3. At the “Select OPTION to schedule or reschedule:” prompt, enter **XU EPCS LOGICAL ACCESS**.
4. At the “...OK? Yes//” prompt, enter **YES**. A ScreenMan dialog is displayed.
5. Tab down to the following fields and enter the values shown:
    - QUEUED TO RUN AT WHAT TIME: **T+1@001** (which means start running it tomorrow at 12:01)
    - RESCHEDULING FREQUENCY: **1D** (which means run it daily)
6. At the “COMMAND:” prompt, enter **Save**.
7. At the “COMMAND:” prompt, enter **Exit**.

Figure 72: DEA ePCS: Task Changes to DEA Prescribing Privileges Report Option: TaskMan schedule setup—Sample User Entries



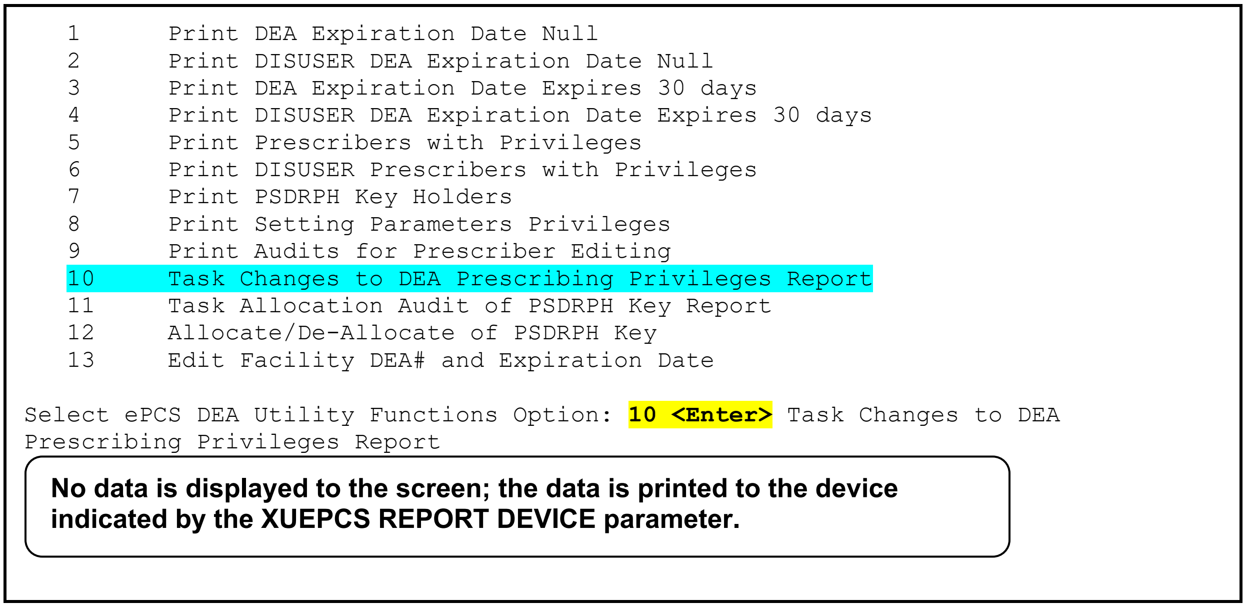
```
          Device Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
   FM1    VA FileMan ...
   JL     Consolidated Practitioner’s Menu ...
          Application Utilities ...
          Capacity Planning ...
          Manage Mailman ...
          Menu Management ...
          Verifier Tools Menu ...

Select Systems Manager Menu Option: TASK <Enter> man Management

          Schedule/Unschedule Options
          One-time Option Queue
          Taskman Management Utilities ...
          List Tasks
          Dequeue Tasks
          Requeue Tasks
          Delete Tasks
          Print Options that are Scheduled to run
   TU     TASK UTILITY
   VPD    Cleanup Task List
          Print Options Recommended for Queueing

Select Taskman Management Option: SCHED <Enter> ule/Unschedule Options

Select OPTION to schedule or reschedule: XU EPCS LOGICAL ACCESS <Enter>     Task
Changes to DEA Prescribing Privileges Report
         ...OK? Yes// <Enter> (Yes)
      (R)
                          Edit Option Schedule
    Option Name: XU EPCS LOGICAL ACCESS
    Menu Text: Task Changes to DEA Prescribing                TASK ID:
  _____________________________________________________________________________


  QUEUED TO RUN AT WHAT TIME: T+1@001

DEVICE FOR QUEUED JOB OUTPUT:

 QUEUED TO RUN ON VOLUME SET:

      RESCHEDULING FREQUENCY: 1D

             TASK PARAMETERS:

            SPECIAL QUEUEING:

________________________________________________________________________________
Exit     Save     Next Page     Refresh
```

Tab to the fields indicated and enter the values shown.



```
Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: SAVE                                   Press <PF1>H for help    Insert
.
.
.
________________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: EXIT                                    Press <PF1>H for help    Insert

Select OPTION to schedule or reschedule:
```

Figure 73: DEA ePCS: Task Changes to DEA Prescribing Privileges Report Option—Sample User Entries (No Report Displays)



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 10 <Enter> Task Changes to DEA
Prescribing Privileges Report
```

No data is displayed to the screen; the data is printed to the device indicated by the XUEPCS REPORT DEVICE parameter.

6.4.12 Task Allocation Audit of PSDRPH Key Report Option

CAUTION: Verify that the XUEPCS REPORT DEVICE parameter has been set before using this option.

To set the parameter, see the "Set the XUEPCS REPORT DEVICE Parameter" section.

The **Task Allocation Audit of PSDRPH Key Report** [XU EPCS PSDRPH AUDIT] option prints the allocation of the PSDRPH security key audit report to a device previously selected during setup (i.e., XUEPCS REPORT DEVICE parameter).

This option should be scheduled to run on a daily basis via TaskMan. The option only prints data from the *previous* day and with *data that has been modified*. The data is retrieved from the XUEPCS PSDRPH AUDIT (#8991.7) file.

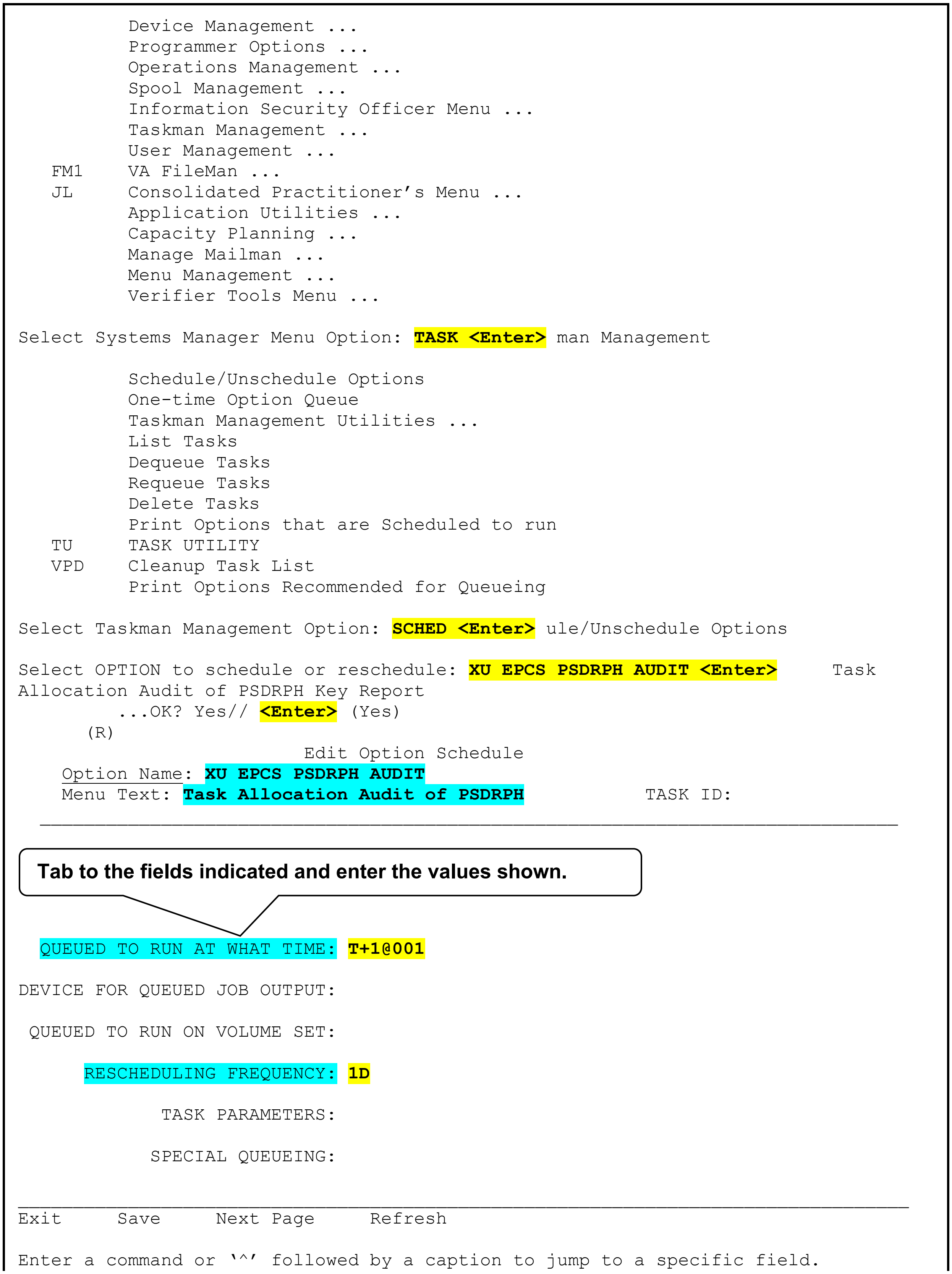
NOTE: No data is displayed to the screen; the data is printed to the device indicated by the XUEPCS REPORT DEVICE parameter.

NOTE: This option was released with Kernel patch XU*8.0*580.

To schedule the option to run daily using TaskMan, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **Taskman Management** [XUTM MGR] option.
2. At the "Select Taskman Management Option:" prompt, select the **Schedule/Unschedule Options** [XUTM SCHEDULE] option.
3. At the "Select OPTION to schedule or reschedule:" prompt, enter **XU EPCS PSDRPH AUDIT**.
4. At the "...OK? Yes//" prompt, enter **YES**. A ScreenMan dialog is displayed.
5. Tab down to the following fields and enter the values shown:
   - QUEUED TO RUN AT WHAT TIME: **T+1@001** (which means start running it tomorrow at 12:01)
   - RESCHEDULING FREQUENCY: **1D** (which means run it daily)
6. At the "COMMAND:" prompt, enter **Save**.
7. At the "COMMAND:" prompt, enter **Exit**.

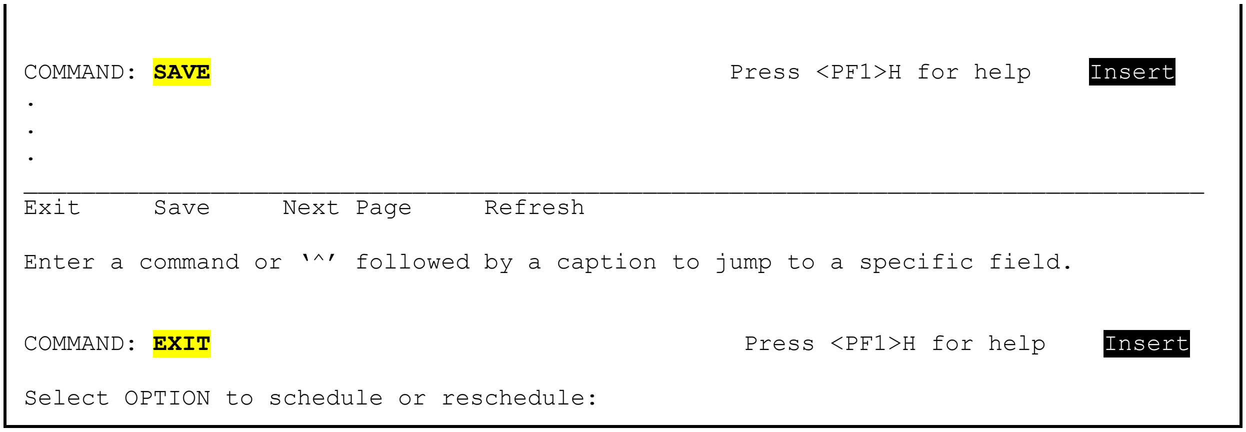
Figure 74: DEA ePCS: Task Allocation Audit of PSDRPH Key Report Option: TaskMan Schedule Setup—Sample User Entries



```
          Device Management ...
          Programmer Options ...
          Operations Management ...
          Spool Management ...
          Information Security Officer Menu ...
          Taskman Management ...
          User Management ...
   FM1    VA FileMan ...
   JL     Consolidated Practitioner's Menu ...
          Application Utilities ...
          Capacity Planning ...
          Manage Mailman ...
          Menu Management ...
          Verifier Tools Menu ...

Select Systems Manager Menu Option: TASK <Enter> man Management

          Schedule/Unschedule Options
          One-time Option Queue
          Taskman Management Utilities ...
          List Tasks
          Dequeue Tasks
          Requeue Tasks
          Delete Tasks
          Print Options that are Scheduled to run
   TU     TASK UTILITY
   VPD    Cleanup Task List
          Print Options Recommended for Queueing

Select Taskman Management Option: SCHED <Enter> ule/Unschedule Options

Select OPTION to schedule or reschedule: XU EPCS PSDRPH AUDIT <Enter>     Task
Allocation Audit of PSDRPH Key Report
         ...OK? Yes// <Enter>  (Yes)
      (R)
                          Edit Option Schedule
    Option Name: XU EPCS PSDRPH AUDIT
    Menu Text: Task Allocation Audit of PSDRPH          TASK ID:
  ______________________________________________________________________________
```

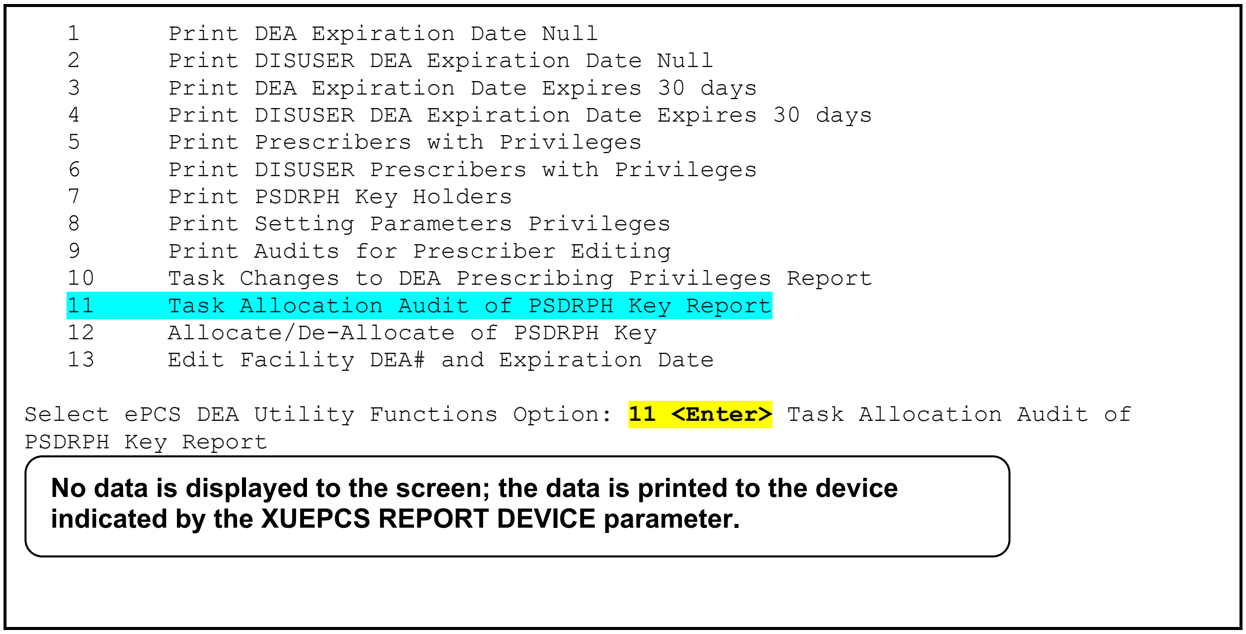
Tab to the fields indicated and enter the values shown.

```
  QUEUED TO RUN AT WHAT TIME: T+1@001

DEVICE FOR QUEUED JOB OUTPUT:

 QUEUED TO RUN ON VOLUME SET:

      RESCHEDULING FREQUENCY: 1D

             TASK PARAMETERS:

            SPECIAL QUEUEING:

_________________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.
```



```
COMMAND: SAVE                                     Press <PF1>H for help    Insert
.
.
.
_________________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: EXIT                                      Press <PF1>H for help    Insert

Select OPTION to schedule or reschedule:
```

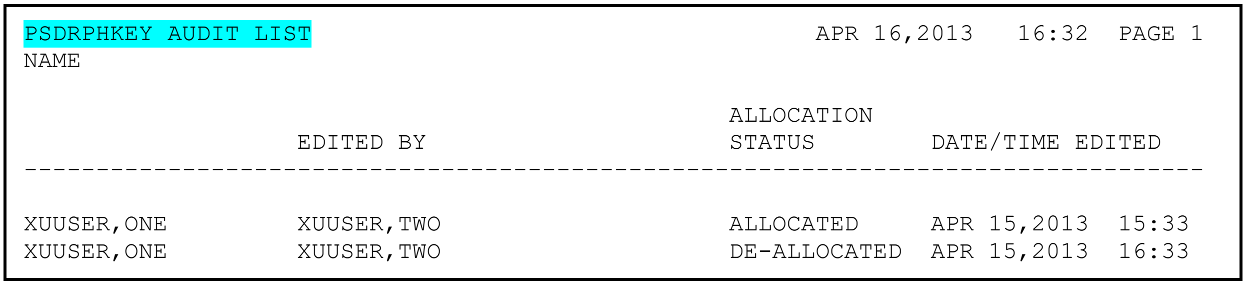
Figure 75: DEA ePCS: Task Allocation Audit of PSDRPH Key Report Option—Sample User Entries (No Report Displays)



```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 11 <Enter> Task Allocation Audit of
PSDRPH Key Report
```

No data is displayed to the screen; the data is printed to the device indicated by the XUEPCS REPORT DEVICE parameter.

Figure 76: DEA ePCS: Task Allocation Audit of PSDRPH Key Report Option—Sample Report Printed to Device Entered into the XUEPCS REPORT DEVICE Parameter



```
PSDRPHKEY AUDIT LIST                                    APR 16,2013   16:32   PAGE 1
NAME

                                                  ALLOCATION
                   EDITED BY                      STATUS        DATE/TIME EDITED
---------------------------------------------------------------------------------

XUUSER,ONE         XUUSER,TWO                     ALLOCATED     APR 15,2013  15:33
XUUSER,ONE         XUUSER,TWO                     DE-ALLOCATED  APR 15,2013  16:33
```

6.4.13 Allocate/De-Allocate of PSDRPH Key Option

The **Allocate/De-Allocate of PSDRPH Key** [XU EPCS PSDRPH KEY] option allocates or de-allocates the PSDRPH security key.

NOTE: All user security keys are stored in the KEYS (#51) Multiple field in the NEW PERSON (#200) file.

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 77: DEA ePCS: Allocate/De-Allocate of PSDRPH Key Option: *Allocating* PSDRPH—Sample User Entries

```
   1      Print DEA Expiration Date Null
   2      Print DISUSER DEA Expiration Date Null
   3      Print DEA Expiration Date Expires 30 days
   4      Print DISUSER DEA Expiration Date Expires 30 days
   5      Print Prescribers with Privileges
   6      Print DISUSER Prescribers with Privileges
   7      Print PSDRPH Key Holders
   8      Print Setting Parameters Privileges
   9      Print Audits for Prescriber Editing
   10     Task Changes to DEA Prescribing Privileges Report
   11     Task Allocation Audit of PSDRPH Key Report
   12     Allocate/De-Allocate of PSDRPH Key
   13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 12 <Enter> Allocate/De-Allocate of PSDRPH
Key
Enter User Name: XUSER
     1   XUUSER,ONE        OX
     2   XUUSER,TWO        TX     192     SYSTEMS ANALYST
     3   XUUSER,THREE B        TBX
     4   XUUSER,FOUR        FX
     5   XUUSER,FIVE A       FAX
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-5: 2 <Enter> XUUSER,TWO     TX     192     SYSTEMS ANALYST
Allocate PSDRPH for XUUSER,TWO? YES// <Enter>
```

Figure 78: DEA ePCS: Allocate/De-Allocate of PSDRPH Key Option: *De-allocating* PSDRPH—Sample User Entries

```
Select ePCS DEA Utility Functions Option: 12 <Enter> Allocate/De-Allocate of PSDRPH
Key
Enter User Name: XUUSER,TWO <Enter>  XUUSER,TWO     TX     192     SYSTEMS ANALYST
De-allocate PSDRPH for XUUSER,TWO? YES// <Enter>
```

REF: To review the audit history of the allocation and de-allocation of the PSDRPH security key, see the sample report generated from the **Task Allocation Audit of PSDRPH Key Report** [XU EPCS PSDRPH AUDIT] option in Figure 76.

6.4.14 Edit Facility DEA# and Expiration Date Option

The **Edit Facility DEA# and Expiration Date** [XU EPCS EDIT DEA# AND XDATE] option edits the FACILITY DEA NUMBER (#52) and FACILITY DEA EXPIRATION DATE (#52.1) fields in the INSTITUTION (#4) file.

NOTE: This option was released with Kernel patch XU*8.0*580.

Figure 79: DEA ePCS: Edit Facility DEA# and Expiration Date Option—Sample User Entries

```
    1      Print DEA Expiration Date Null
    2      Print DISUSER DEA Expiration Date Null
    3      Print DEA Expiration Date Expires 30 days
    4      Print DISUSER DEA Expiration Date Expires 30 days
    5      Print Prescribers with Privileges
    6      Print DISUSER Prescribers with Privileges
    7      Print PSDRPH Key Holders
    8      Print Setting Parameters Privileges
    9      Print Audits for Prescriber Editing
    10     Task Changes to DEA Prescribing Privileges Report
    11     Task Allocation Audit of PSDRPH Key Report
    12     Allocate/De-Allocate of PSDRPH Key
    13     Edit Facility DEA# and Expiration Date

Select ePCS DEA Utility Functions Option: 13 <Enter> Edit Facility DEA# and
Expiration Date

Select INSTITUTION NAME: SAN FRANCISCO
     1   SAN FRANCISCO         CA  VAMC      662
     2   SAN FRANCISCO         CA  VCSFO     782
     3   SAN FRANCISCO         CA  NC        903
     4   SAN FRANCISCO-OPT     CA
     5   SAN FRANCISCO-RO      CA  RO        343
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-5: 1 <Enter> SAN FRANCISCO   CA  VAMC      662
FACILITY DEA NUMBER: BB1234563// ?
     Answer with a DEA ID, must be 9 characters in length
FACILITY DEA NUMBER: BB1234563// <Enter>
FACILITY DEA EXPIRATION DATE: SEP 9,2011// <Enter>

Select INSTITUTION NAME:
```

6.4.15 ePCS Edit Prescriber Data Option

The **ePCS Edit Prescriber Data** [XU EPCS EDIT DATA] option is a Broker-type context option that is given to those individuals who are permitted to edit the data related to e-prescribing of controlled substances.

This option is locked with the XUEPCSEDIT security key.

NOTE: This option was released with Kernel patch XU*8.0*580.

6.4.16 ePCS Set SAN from PIV Card Option

The **ePCS Set SAN from PIV Card** [XUSSPKI UPN SET] option is a Broker-type context option that sets the SUBJECT ALTERNATIVE NAME (#501.2) field (a.k.a. SAN field or USER PRINCIPLE NAME) in the NEW PERSON (#200) file from the Personal Identification Verification (PIV) Smart Card. This is used with the DEA ePCS electronic signature (e-sig) to be sure the correct certificate is selected from the PIV card.

NOTE: This option only needs to be run once for a user at a site.

NOTE: This option was released with Kernel patch XU*8.0*580.

6.4.16.1 XUSSPKI SAN Bulletin

Released with Kernel patch XU*8.0*580, the XUSSPKI SAN bulletin is sent when the SUBJECT ALTERNATIVE NAME (#501.2) field in the NEW PERSON (#200) file has been changed or deleted. The bulletin is sent to users holding the PSDMGR security key.

- **Subject:** “Subject Alternative Name” field
- **Message:** The “Subject Alternative Name” field in New Person File (#200) has been changed or deleted for: **|3|**

  Before: |1|

  After: |2|

  NOTE: If this value is **NULL**, the field was deleted!

- Parameters:
  - **|1|**—Old value before changed or deleted.
  - **|2|**—New value. If **NULL**, value was deleted.
  - **|3|**—Name of the user.

6.5 Prescription Validation and Verification Process—PKIServer.exe Application

The **PKIServer.exe** is an application that runs as a service application to handle verification of prescriptions that have been entered using the electronic prescribing of controlled substances (ePCS) in the Computerized Patient Record System (CPRS) application. The **PKIServer.exe** application itself is written in the Delphi language and uses the cryptographic APIs within the Windows operating system.

REF: For more information on cryptographic functions, see the "Windows Authentication and Cryptographic Operations" section.

NOTE: The VA was the original test site (at the Hines VAMC) for ePCS for the DEA starting in 2002 with code in CPRS for this purpose. That test site has continued to use this functionality (and the functionality has been in CPRS) until the current time. The DEA has now come up with the final rules for the use of ePCS and the version of CPRS that is currently in testing moves the functionality to meet the final regulations and expands its use to all sites instead of the single Hines site.

There is code within CPRS that handles the following:

- Cryptographic functionalities involved in verifying the provider's pin value for the PIV card (the original testing used cards provided by DEA).

  REF: For more information on cryptographic functions, see the "Windows Authentication and Cryptographic Operations" section.

- Validation of the PIV card with respect to expiration or revocation.

  REF: For more information on revoked VA PIV cards, see the "PIV Card Validation—Revocation Server" section.

- Creation of the hash for the aggregate prescription data and signing of that hash. The signed hash is created so that it contains a copy of the signing certificate as well.

At the time that the pharmacist goes to fill the prescription there are requirements that the prescription be validated to insure that there have been no changes to the data associated with the prescription before it is filled. The pharmacist works within the VistA roll-and-scroll environment, which does *not* offer the capabilities required to provide the cryptographic checks necessary.

To validate the prescription using cryptographic checks, the system performs the following procedure:

1. VistA Pharmacy code passes the current data associated with the prescription and the signed hash value via Kernel utilities to a server location identified by the PKI SERVER (#53.1) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file. There can be up to three IP addresses separated by caret characters (^) in this field. This connects the VistA server to the PKIServer service (identified in the services functionality as PKI_Verify_Service).
2. PKIServer takes the input data and extracts the signing certificate and original hash from the signed hash.
3. PKIServer creates a hash of the current data passed in for the prescription.
4. PKIServer compares the two hashes:
   - **Hashes match**—If the two hashes match, indicating no change in the data, the PKIServer then checks whether the certificate has been revoked (see Step 5).
   - **Hashes do *not* match**—If any changes have occurred in the data currently associated with the prescription, the two hashes differ:
     a. PKIServer returns a value indicating prescription is returned.
     b. Prescription is voided.

5. PKIServer checks whether the certificate has been revoked:
   - **Active Certificate**—If the hashes match and there is confirmation that the certificate has *not* been revoked, the prescription is approved.
   - **Revoked Certificate**—If the provider's certificate has been revoked, the prescription is voided as well.
   - **Pending Certificate Check**—There may be cases where there are problems in checking the certificate and a return value in this case may indicate that they should wait and check the prescription later.

To meet the DEA requirements, newer, higher level cryptographic methods are required than were previously used in the original Hines testing, and these may require that older server systems be patched to insure that capabilities (e.g., SHA-2) are available. Also, the VA has been moving to use functionality (e.g., Tumbleweed Desktop Validator) to assist in checking certificate statuses, etc. The **PKIServer.exe** application does *not* call these directly; however, if they are available, they are called by the Windows operating system via the cryptographic APIs.

REF: For more information on the **PKIServer.exe** application, see the *DEA e-Prescribing Installation and Setup Guide* located under CPRS on the VDL: VDL CPRS Application Documents

6.6 PIV Card Validation—Revocation Server

The Revocation Server contains a Certificate Revocation List (CRL), which is a list of all revoked VA PIV cards. The distinction is that if a physician prescribes a drug, and then the physician's certificate expires *before* the prescription is filled, it can still be filled, since it was written *before* it expired. If, however, the physician's certificate is revoked, then any orders that have *not* been filled are cancelled and *cannot* be filled. In many cases, certificates are revoked due to a change in affiliation.

To check the CRL to see if a PIV card has been revoked, perform the following procedure:

1. Insert the **PIV card**.
2. Double click on the **ActivClient Agent** to open it.
3. Click on the **My Certificates** icon.
4. Select and double click on one of the certificates.
5. Click on the **Advanced** tab.
6. Scroll down to find and select the **CRL Distribution Points** entry. The CRL is the Certificate Revocation List.
7. Scroll down and see the contents for this entry. You should probably find an entry for the following:
    - one **http:** entry
    - one **ldap:** entry. For example:

        URL=http://cdp1.ssp-strong-id.net/CDP/vauser.crl

8. Copy the **http://** URL address and paste it into a Web browser. It brings up a long list of all of the certificates that have been revoked (as opposed to expired, cancelled, etc.). You should get approximately **30 Megabytes** for the Web page.

    The Tumbleweed Desktop Validator is supposed to assist with this if it is on the desktop, and it updates itself at intervals, so that the call does *not* have to be made to the site for each individual request.

6.7 Windows Authentication and Cryptographic Operations

6.7.1 History

The VA's attempt to use Microsoft® Windows-level authentication to access VistA accounts using a secure intermediary authentication server was set to be released in the late 1990s via the Enterprise Single Sign-On (ESSO) patch. During that time the Office of Cyber Security informed the VA that they had a better way and would implement it within six months. Subsequently, the VA stopped the release of the ESSO patch, but nothing more happened with regard to Microsoft® Windows level authentication.

In 2015, the VA began development of Single Sign-On Internal (SSOi) using Identity and Access Management (IAM) Secure Token Service (STS) to enable 2-Factor Authentication (2FA) of VA employees into VistA. Kernel patches XU*8.0*655 and XU*8.0*659 enable authentication into VistA using an STS token obtained from IAM. Single Sign-On External (SSOe) authentication of veterans and *non*-VA VistA users is currently in development.

6.7.2 Current Capabilities

VistA Kernel provides the mechanism to authenticate a user with an STS token obtained from IAM. VistA does *not* do direct authentication of a user via a PIV card or similar means. Authentication via PIV card is delegated to IAM. VistA validates a PKI certificate and digital signature from IAM to secure the delegated authentication process. This process is currently enabled for Remote Procedure Call (RPC) Broker and VistALink applications.

CPRS v30 is capable of handling the electronic prescribing of controlled substances, but all of the cryptographic operations are handled via the client workstation (for the signing of the prescription). This is before the data is passed to the VistA server, along with a copy of the signed hash generated based on the data for the prescription. At the time of filling of the prescription by the VA pharmacist, the data for the prescription along with a copy of the signed hash is transferred by VistA to a PKIService application. This PKIService application runs on a separate server or workstation for verification that the data associated with the prescription has *not* changed. It compares the original hash value with one created based on the current data.

REF: For more information on the PKIService verification process, see the "Prescription Validation and Verification Process—PKIServer.exe Application" section.

6.7.3 Future Capabilities

Terminal access (roll-and-scroll) VistA 2-Factor Authentication (2FA) is currently in development. This process will require a script within the terminal emulator software to call IAM to authenticate the user via PIV or similar means, and then send the returned STS token to VistA for authentication and identification of the user. Single Sign-On External (SSOe) is currently in development to use 2-Factor Authentication (2FA) to authenticate and identify external (*non*-VistA) users to obtain or edit data within VistA. External users include:

- Veterans
- Department of Defense (DoD) users
- *Non*-VA providers who require access to veteran data.

External users will be required to authenticate with IAM and use the returned STS token to authenticate and identify the user within VistA. Since these users might *not* be currently "known" to VistA, a means of role-based authorization is required to provision the users on-the-fly and restrict their access to specific data based upon their role. Role-based authorization for external VistA users has yet to be developed.

II. Menu Manager

7 Menu Manager: User Interface

Kernel’s menu system presents menu options within VistA software in a standard fashion. Once you become familiar with using the menu system in one application, using other applications is easier, since the same rules apply.

7.1 Navigating Kernel’s Menus

When you successfully sign into the computer system, Menu Manager presents your primary menu options. Your primary menu is the top-level menu assigned to you by the system administrators. Most options that are available to you are available from your primary menu, or from a submenu attached to your primary menu.

The menu system prompts you with a “Select (menu name) Option:” prompt. For example, in a menu named Billing, Menu Manager would prompt you with “Select Billing Option:”. You can navigate through the menu system by responding to this prompt in different ways, which are described in this section.

You can enter question marks to see option choices and obtain online help. You can enter an option’s synonym or the first few letters of its menu text, using upper or lowercase, to select the option. You can also enter a caret (^) along with the option specification (option menu text or synonym) to jump to the destination option rather than traversing the menu pathways step-by-step.

7.1.1 Choosing Options

You can choose an option from your current menu at the select prompt. Choosing the option launches the software application associated with the option. To choose an option, type in the first few letters of the option as it is displayed and press the **<Enter>** key. If multiple options match those first few characters you are presented with a list of matching options from which you can choose the specific option you want to run. If the option is another menu, indicated by trailing ellipses (**...**), it becomes the current menu, and so on down the menu pathway.

To come back up the menu pathway, press **<Enter>** at the select prompt. Each time you press **<Enter>**, Menu Manager returns you to the next higher menu level, until you reach your highest menu, the primary menu. If you press **<Enter>** at the primary menu, Menu Manager asks if you want to halt your session. If you answer **YES**, your Kernel session is ended.

7.1.2 Listing Options

When you enter a menu, the items may or may *not* be displayed automatically, based on whether you have AUTO MENU turned on. The AUTO MENU feature, as described in the "Signon/Security: User Interface" section, is a flag that controls the menu display. If you do *not* have a setting specified for AUTO MENU, the site parameter default is used. Often, to save system resources, the site parameter can be set to disable automatic display. In this case, to display menu items, simply enter a single question mark (**?**), as shown in Figure 80:

Figure 80: One Question Mark (?) Help—Sample User Dialog

```
Select Any Level Menu Option: ?

          First Item
          Second Item
          Third Item of Menu Choices ...
          Fourth Item

Enter ?? for more options, ??? for brief descriptions, ?OPTION for help text.

Select Any Level Menu Option:
```

7.1.3 Displaying Option Help

To obtain a lengthier description of an individual option, enter a single question mark (**?**), and the first few letters of the option name. If there is an extended description of the option, or a help frame describing the option, they are displayed, as shown in Figure 81:

Figure 81: Using ?Option to Get Help on a Named Option—Sample User Dialog

```
Select User's Toolbox Option: ?

          Display User Characteristics
          Edit User Characteristics
          Electronic Signature Code Edit
          Menu Templates...
          Spooler Menu...
          TaskMan User
          User Help

Select User's Toolbox Option: ?DISPLAY



'Display User Characteristics'     Option name: XUUSERDISP
    Display the user's name, location, and characteristics

 **> Press 'RETURN' to continue, '^' to stop: <Enter>

Select User's Toolbox Option:
```

7.1.4 Listing Secondary and Common Options

At any select prompt you can enter two question marks (**??**) to see options on the **Secondary** menu and **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu), as well as options available on the current branch of your menu tree.

The **Secondary** and **Common** menus contain options that you can select at any location in the menu system:

- Options on the **Secondary** menu are typically created by your system manager.
- Options on the **Common** menu are standard Kernel options available from anywhere in the menu system.
- Options on the current menu, on the other hand, can only be directly selected while that menu is the current menu.

The two-question-mark display shows the option's synonym (a short abbreviation), if one exists. You can select an option by its synonym as well as by its full name. On the same line, it lists the option's full name followed by the formal option name in capital letters enclosed in square brackets. (The name is the .01 field of the OPTION [#19] file.) It also shows any option restrictions such as:

- Out-of-Order
- Locked
- Prohibited times

]

Figure 82: Two Question Marks (??) Help—Listing Primary, Secondary, and Common Menu Options

```
Select Systems Manager Menu Option: ??

   FM     VA FileMan ...                                             [DIUSER]
          Core Applications ...                                      [XUCORE]
          Device Management ...                                       [XUTIO]
             **> Locked with XUPROG
          Information Security Officer Menu ...                       [XUSPY]
          Manage Mailman ...                                          [XMMGR]
          Menu Management ...                                       [XUMAINT]
          Operations Management ...                               [XUSITEMGR]
          Programmer Options ...                                     [XUPROG]
             **> Locked with XUPROG
          Spool Management ...                                   [XU-SPL-MGR]
          Taskman Management ...                                   [XUTM MGR]
          User Management ...                                         [XUSER]

You can also select a secondary option:

   OUT    Equipment Checked Out to Myself                    [A6A EQUIP USER]
   PAID   SIGN INTO MARTINEZ VIA TELNET, TYPE DUSER            [A6A USE PAID]
   RUM    Capacity Planning ...                                  [XTCM MAIN]
          ISC OFFICE MENU OPTIONS ...                              [ISCSTAFF]

Or a Common Option:

   KNF    Kernel New Features Help                       [XUVERSIONEW-HELP]
          Halt                                                       [XUHALT]
          Continue                                               [XUCONTINUE]
          Restart Session                                          [XURELOG]
   MM     MailMan Menu ...                                           [XMUSER]
   NPI    PROVIDER NPI SELF ENTRY                [XUS NPI PROVIDER SELF ENTRY]
   TBOX   User’s Toolbox ...                                     [XUSERTOOLS]
   VA     View Alerts                                               [XQALERT]
          Time                                                       [XUTIME]
          Where am I?                                            [XUSERWHERE]
```

7.1.5 Displaying Option Descriptions

Entering three question marks (**???**) at any select prompt displays option descriptions (from a word-processing-type field in the OPTION [#19] file). If entered at the select prompt for a menu within the primary tree, the top-level options are described; then you are prompted whether you want to see descriptions for **Secondary** or **Common** options.

Figure 83: Three Question Marks (???) Help—Sample User Dialog

```
Select Spooler Menu Option: ???

'Allow other users access to spool documents'     Option name: XU-SPL-ALLOW
     This option edits the 'OTHER AUTHORIZED USERS' field of the SPOOL
     DOCUMENT file to allow other users access to a spool document.

'Delete A Spool Document'     Option name: XU-SPL-DELETE
  **> Extended help available.  Type "?Delete" to see it.
     Delete a spool document from the spool document file and delete the
     associated message if they are still linked.

'List Spool Documents'     Option name: XU-SPL-LIST
  **> Extended help available.  Type "?List" to see it.
     This option lists entries in the spool document file.

'Make spool document into a mail message'     Option name: XU-SPL-MAIL
  **> Extended help available.  Type "?Make" to see it.
     This option will take a spool document and post it as a mailman
     message to the user's IN basket.  This doesn't move the data at all
     but does decrease the number of lines charged to the user.

  **> Press 'RETURN' to continue, '^' to stop, or '?[option text]' for more
                         help: <Enter>

'Print A Spool Document'     Option name: XU-SPL-PRINT
  **> Extended help available.  Type "?Print" to see it.
     This allows the printing of a document that has been spooled.


  Shall I show you your secondary menus too? No// <Enter>
  Would you like to see the Common Options? No// <Enter>


Select Spooler Menu Option:
```

You should be ready to use three question marks (**???**) to learn more about unfamiliar options (e.g., options distributed in a new software release).

7.1.6 Jumping to Options—”Up-arrow Jump”)

The pathways of the **Primary**, **Secondary**, and **Common**] menus have tree-like structures. You can step up or down the pathways to reach your destination or invoke the menu system’s “Up-arrow Jump” feature as a shortcut. To jump to an option, enter a caret (^) before the option specification (the option’s menu text or synonym in upper- or lowercase letters). You only need to enter the first few characters needed to uniquely identify the option. You can use the option’s synonym to limit ambiguity, especially if the synonym is distinct from other synonyms or menu texts.

Figure 84: Using the “Up-arrow Jump”—Sample User Dialog

```
Select Systems Manager Menu Option: ^INTRO <Enter> ductory text edit
```

The menu system carries out the necessary footwork to reach the desired option. If, along the way, there are pathway restrictions (e.g., locks or prohibited times), access to the option is denied, just as when stepping to an option. If a match is found within the primary or secondary menus, that option is executed.

NOTE: The menu system does *not* search the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu) if it can find a match in the primary or secondary menus.

If the menu system finds *more than one* matching option on *t*he **Primary**, **Secondary**, or **Common** menu tree, the menu system presents a list of matching choices. Entering a caret (^) followed by a question mark (**?**) displays all of the options available to you.

Figure 85: List of Choices—Sample User Dialog

```
Select Systems Manager Menu Option: ^LIST NAMES

    1    List Namespaces   [XUZ NAMESPACES]
    2    List Namespaces   [ZZ NAMESPACE LIST]

Type ‘^’ to stop, or choose a number from 1 to 2 :
```

System administrators should assign “shallow” secondary menus to facilitate menu jumping. When a jump is requested, the menu system searches all the way through the primary as well as the secondary, looking for a match. Users are inconvenienced and system resources are consumed if secondary menus are “deep” in terms of their hierarchical tree-like structure.

You may occasionally find jumping disabled; when you try to jump, you may get a message that quick access is temporarily disabled. Jumping stays disabled until the needed menu trees are rebuilt.

7.1.7 Jumping to Options—”Rubber-band Jump”

The menu system’s jump feature includes the ability to jump out to a destination option and then back again, something like the motion of a rubber band. The syntax for the “Rubber-band Jump” request is the use of a double caret (^^) followed by the usual option specification. For example:

Figure 86: “Rubber-band Jump”—Sample User Dialog

```
Select Systems Manager Menu Option: ^^TASKMAN USER
```

As with the single “Up-arrow Jump” (^), restrictions along the menu pathways are checked.

If you enter two carets (^^) without a following option specification/name, you are returned to the primary menu. This technique is a quick way for you to “go home” to the menu that is displayed at signon; it is called the “Go-home Jump.”

CAUTION: It is important to note that when you invoke the “Rubber-band Jump,” there is no attempt to protect variables that can be SET or KILLed, via Entry or Exit Actions, as you jump through the menu tree. Thus, the “Rubber-band Jump” can be inappropriate under certain circumstances, since it could cause significant alteration of your environment.

7.1.8 Common Menu

The **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu) is designed as a collection of options that are available to all users. The standard **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu items are:

- **User’s Toolbox** [XUSERTOOLS]: As described in the “User’s Toolbox Menu” section in the “Signon/Security: User Interface” section, the User’s Toolbox is a menu containing options that allow users to control some aspects of their computing environment.
- System Logout Options:
    - **Halt** [XUHALT]
    - **Continue** [XUCONTINUE]
    - **Restart Session** [XURELOG]

REF: These three options offer different ways to log out of the system as described in the “Signon/Security: User Interface” section.

- **View Alerts** [XQALERT]**:** As described in the “Alerts” and “Signon/Security: User Interface” sections, this option lets you process alerts.
- **Time** [XUTIME]: This option simply displays the date and time.
- **Where am I?** [XUSERWHERE]: This option lists information identifying what computer system you are signed into (e.g., UCI, Volume Set, Node, and Device).

7.1.8.1 Selecting Common Options with the Double Quote

Since **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu options (aka **Common** menu) are intended to be readily accessible, there is a shortcut method to reach them. While you could use an “Up-arrow Jump,” it is quicker to enter a quotation mark followed by the option specification (e.g., name, synonym). Figure 87 selects the **User’s Toolbox** [XUSERTOOLS] menu from the **Common** menu via its synonym, **TBOX**:

Figure 87: Selecting Common Options via the Double Quote—User’s Toolbox Menu Option

```
Select Sample Menu Option: “TBOX

          Display User Characteristics
          Edit User Characteristics
          Electronic Signature code Edit
          Menu Templates ...
          Spooler Menu ...
          TaskMan User
          User Help

Select User’s Toolbox Option:
```

7.2 Menu Templates Option

Menu templates are like scripts. You can use them to execute a fixed series of options, in sequence. Tools for creating, deleting, listing, and renaming templates are options on the Menu Templates menu, part of the **User’s Toolbox (TBOX)** [XUSERTOOLS] menu:

Figure 88: Menu Templates Option

```
Select Menu Templates Option: ?

          Create a new menu template
          Delete a Menu Template
          List all Menu Templates
          Rename a menu template
          Show all options in a Menu Template

Select Menu Templates Option:
```

When you create a MENU template, you are prompted for a series of options that lead to a final non-menu (i.e., executable) destination option. Once you choose one *non*-menu option to be executed, you can navigate to other options and choose them to be executed as well, if you wish.

When you have selected each executable option to be part of the template, enter a plus sign (+) to store the sequence of options. You are asked to confirm the sequence of options in the template, and then to give the template a name.

To invoke the template, simply enter a left square bracket followed by the template name, as shown in Figure 89:

Figure 89: Invoking a Template—Sample User Dialog

```
Select Option: [MYTEMPLATE

Loading MYTEMPLATE...
```

The template then executes each option that is part of the template, in the same order as the options were selected for the template.

MENU templates are stored in the MENU TEMPLATE (#19.8) Multiple field of the NEW PERSON (#200) file, so you can use any name for MENU templates. If your MENU template points to options that are subsequently removed from the OPTION (#19) file, you receive a message that the MENU template no longer functions properly and needs to be deleted or rebuilt.

Use menu jumping (i.e., the “Up-arrow Jump”) when you want to jump immediately to an option. Use MENU templates when you have a series of options that you need to run in the same order repeatedly, over a period of time.

7.2.1 LOGIN Menu Template

Beginning with Kernel 8.0, you can have a MENU template execute automatically, on your first signon of the day. If you have a MENU template named LOGIN (all uppercase), the MENU template is executed on your first signon of the day. So if you have a series of options you execute on your first signon every day, an easy way to execute them is to create a MENU template; store the series of options in the template; and name the template LOGIN.

7.3 Summary

Once you learn how to navigate Kernel’s menu tree, you can use some of Menu Manager’s additional features to help increase your productivity in the VistA computer system. These features include:

- “Up-arrow Jump”.
- “Rubber-band Jump”.
- Using three question marks (**???**) to obtain online option help.
- Using MENU templates as scripts.

8 Menu Manager: System Management

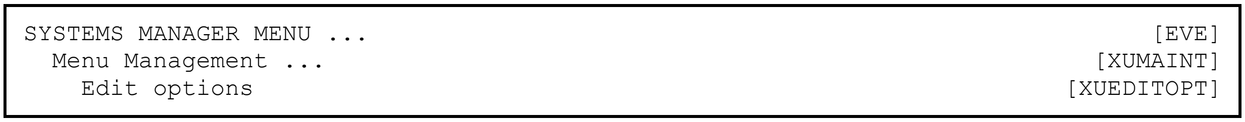
Menu Manager is built around options, which are entries in the OPTION (#19) file. There are several types of options:

- **Menus**—Options with subentries in the MENU (#10; item) Multiple field.
- **Multiples**—Options that point back to the OPTION (#19) file itself.
- **Plugins**—Options that are designed as items that plug into the MENU (#10, item) Multiple field of a menu-type option.

Kernel provides a number of tools to create and manage menus and options.

8.1 Creating Menus and Options

Figure 90: Edit Options Option



```
SYSTEMS MANAGER MENU ...                                          [EVE]
  Menu Management ...                                         [XUMAINT]
    Edit options                                            [XUEDITOPT]
```

One task system administrators perform frequently is defining local primary menus that are appropriate for their users. This task of menu creation is accomplished by grouping exported menus from various software applications together on a new master menu. You can use **Edit options** [XUEDITOPT], on the **Menu Management** [XUMAINT] menu, to define a new menu if **READ**, **WRITE**, and **LAYGO** access to the OPTION (#19) file has been granted (either through the FILE MANAGER ACCESS CODE [#3] field or through the File Access Security system if that is enabled). Only a few fields need to be defined, as shown in Figure 91. The new menu can then be assigned to a user, as described in the "Signon/Security: User Interface" section, with one of several options.

Figure 91: Defining Local Primary Menus (System Administrators)—Sample User Dialog

```
Select OPTION to edit: ZZSTAFF MENU
  Located in the Z (Local) namespace.
  ARE YOU ADDING 'ZZSTAFF MENU' AS A NEW OPTION (THE 721ST)? Y <Enter> (YES)
   OPTION MENU TEXT: STAFF MENU
NAME: ZZSTAFF MENU// <Enter>
MENU TEXT: Staff Menu// <Enter>
PACKAGE: <Enter>
OUT OF ORDER MESSAGE: <Enter>
LOCK: <Enter>
REVERSE/NEGATIVE LOCK: <Enter>
DESCRIPTION:
  1>This is the primary menu for staff members.
  2><Enter>
EDIT Option: <Enter>
TYPE: MENU
Select ITEM: XUCORE <Enter>    Core Applications
  ARE YOU ADDING 'XUCORE' AS A NEW MENU (THE 1ST FOR THIS OPTION)? Y <Enter> (YES)
   MENU SYNONYM: <Enter>
  SYNONYM: <Enter>
  DISPLAY ORDER: 10
Select ITEM: XUSPY <Enter>   System Security
  ARE YOU ADDING 'XUSPY' AS A NEW MENU (THE 2ND FOR THIS OPTION)? Y <Enter> (YES)
   MENU SYNONYM: <Enter>
  SYNONYM: <Enter>
  DISPLAY ORDER: 20
Select ITEM: XT-KERMIT MENU <Enter>    Kermit menu
  ARE YOU ADDING 'XT-KERMIT MENU' AS A NEW MENU (THE 3RD FOR THIS OPTION)?
YES <Enter> (YES)
   MENU SYNONYM: <Enter>
  SYNONYM: <Enter>
  DISPLAY ORDER: 30
Select ITEM: <Enter>
CREATOR: SITE,MANAGER// <Enter>
HELP FRAME: <Enter>
PRIORITY: <Enter>
Select TIMES PROHIBITED: <Enter>
Select TIME PERIOD: <Enter>
RESTRICT DEVICES?: <Enter>
Select PERMITTED DEVICE: <Enter>
```

8.1.1 Option Name and Menu Text

By convention, the formal option name is usually entered in all capital letters. According to namespacing conventions, it *must* begin with a namespace that identifies the associated software. It is the NAME (#.01) field of the OPTION (#19) file. The menu text is what is displayed to the user at the select prompt. Like the words of a heading or title, initial capitalization is used for all words except prepositions and articles, all of which are presented in lowercase. To minimize the number of keystrokes needed to select an option, different first letters should be used for the text of each menu item. Menus should be limited to about seven items, so they all appear together on one screen. The most frequently used items should be presented first.

8.1.2 Synonyms and Display Order

By default, the items on the menu are displayed in alphabetical order by menu text. If any of the items is assigned a synonym, those items are displayed before others lacking synonyms. To facilitate menu jumping, synonyms should ideally be unique; numbers are *not* good choices for synonyms.

To customize the order of the display, each item on the menu can be assigned a Display Order. This field is an option attribute that is presented when using **Edit options** [XUEDITOPT]. When first assigning a number for the display order, you may want to use **10**, **20**, and **30** rather than **1**, **2**, and **3** to permit easier modification in the future if another item needs to be inserted.

8.1.3 PRIORITY

You can set an option's PRIORITY field to set a run priority for an option. Experimentation is needed to determine the effect of priority settings.

8.1.4 HELP FRAME

You can specify a help frame for an option. The help frame is displayed if, at the "Select..." menu prompt, the user enters ?OPTION (where OPTION is the name of an option).

8.1.5 DISPLAY OPTION

If AUTO MENU (#200.06) is in effect for a user, the items on that user's current menu are always displayed. A problem can arise when, if an option displays output and then quits, AUTO MENU's automatic display of menu options scrolls the output off the screen. Since the AUTO MENU display usually scrolls the option's output off the screen faster than the user can read the output, it can effectively render the option unusable. You can avoid this problem by setting the option's DISPLAY OPTION (#11) field in the OPTION (#19) file to **YES**. If set to **YES** and the user has AUTO MENU turned on, Menu Manager prompts "Press RETURN to continue..." after the option completes, but before displaying the list of menu options. The user then has a chance to review the output before returning to their menu.

REF: For information on other fields in the OPTION (#19) file, including how to create options of a type other than MENU, see the "Menu Manager: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

8.1.6 If the Option Invokes Non-VistA Applications

If you create an option that invokes *non*-VistA applications (e.g., Class III software) include a call to the Device Handler with the code **D HOME^%ZIS** in the EXIT ACTION field of the OPTION (#19) file so that the required **IO** variables is present when leaving these options. Do the same for any other utility that is known to **KILL IO** variables upon exit.

8.1.7 If the Option Should Be Regularly Scheduled

If an option should be regularly scheduled to run through TaskMan, you *must* set its SCHEDULING RECOMMENDED (#209) field in the OPTION (#19) file to **YES**. You are *not* able to use the **Schedule/Unschedule Options** [XUTM SCHEDULE] option to schedule an option unless this field is set to **YES** for the option.

8.1.8 Auditing Option Use

Figure 92: Auditing Menu Options

```
SYSTEM MANAGER MENU...                                                   [EVE]
  System Security...                                                   [XUSPY]
    Audit Features ...                                          [XUAUDIT MENU]
      Maintain System Audit Options ...                        [XUAUDIT MAINT]
        Establish System Audit Parameters                            [XUAUDIT]
        Audited Options Purge                                     [XUOPTPURGE]
      Audit Display ...                                              [XUADISP]
        Option Audit Display                                       [XUOPTDISP]
```

You can establish an audit on options to record every time an option is used. You can do this with the **Establish System Audit Parameters** [XUAUDIT] option, which is in the **Audit Features** [XUAUDIT MENU] menu tree. Simply enter a time to initiate audit and a time to terminate audit. Then enter the specific options you want to audit (you can also choose all options).

Each time a user uses an audited option, an entry is made in the AUDIT LOG FOR OPTIONS (#19.081) file. You can display these entries using the **Option Audit Display** [XUOPTDISP] option. You can purge the AUDIT LOG FOR OPTIONS (#19.081) file with the **Audited Options Purge** [XUOPTPURGE] option.

If Kernel Toolkit is installed at your site, you can also use its **Alpha/Beta Test Option Usage** [XQAB MENU] menu to count the number of times an option is invoked.

REF: For more information, see the Kernel Toolkit documentation and the *Kernel Security Tools Manual*.

8.2 Display Menus and Options Menu

Figure 93: Display Menus and Options Menu

```
SYSTEMS MANAGER MENU ...                                                 [EVE]
  Menu Management ...                                                [XUMAINT]
    List Options by Parents and Use                                   [XUXREF]
    Display Menus and Options                              [XQDISPLAY OPTIONS]
      Abbreviated Menu Diagrams                                   [XUUSERACC2]
      Diagram Menus                                                [XUUSERACC]
      Inquire                                                      [XUINQUIRE]
      Menu Diagrams (with Entry/Exit Actions)                     [XUUSERACC1]
      Print Option File                                              [XUPRINT]
```

Kernel provides a number of options to display and diagram menus and options on the **Display Menus and Options** [XQDISPLAY OPTIONS] menu.

8.2.1 Diagramming Options

To discover the menu tree roots of other software applications and how options and suboptions are related, you can use the menu diagramming options in Table 11:

Table 11: Menu Diagramming Options to Discover Tree Roots and Relationships between Options/Suboptions

Menu	Description
Abbreviated Menu Diagrams	Outlines the menu tree.
Diagram Menus	Outlines the menu tree and shows option attributes (e.g., locks and prohibited times).
Menu Diagrams (with Entry/Exit Actions)	Outlines the menu tree, shows option attributes, and shows entry/exit and header actions as well.

Also, the **List Options by Parents and Use** [XUXREF] option identifies which options have "no parents," and thus, are standalone roots. It also indicates whether options are used as primary menus, secondary menus, or as regularly scheduled tasks.

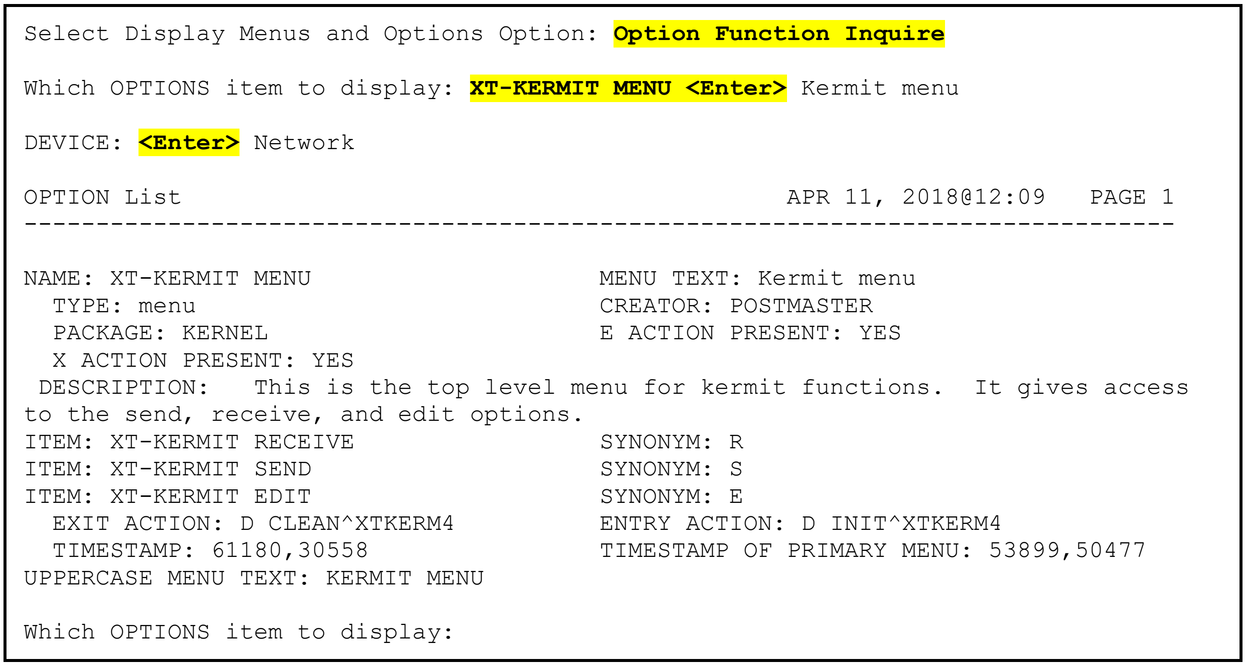
8.2.2 Option Descriptions

To learn more about the options included in a software application, you can use the **Print Option File** [XUPRINT] option (from the **Display Menus and Options** [XQDISPLAY OPTIONS] menu) to print the option description, type, and other information. This listing can be sorted by namespace. For example, to print all the VA FileMan options, you can sort from **DD** to **DI**.

8.2.3 Displaying Options

To display an option, use the Option Function Inquiry [XUINQUIRE] option:

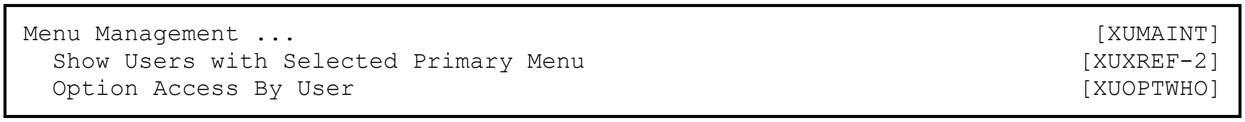
Figure 94: Option Function Inquiry Option—Sample User Dialog and Report



```
Select Display Menus and Options Option: Option Function Inquire

Which OPTIONS item to display: XT-KERMIT MENU <Enter> Kermit menu

DEVICE: <Enter> Network

OPTION List                                          APR 11, 2018@12:09   PAGE 1
--------------------------------------------------------------------------------

NAME: XT-KERMIT MENU                        MENU TEXT: Kermit menu
  TYPE: menu                                CREATOR: POSTMASTER
  PACKAGE: KERNEL                           E ACTION PRESENT: YES
  X ACTION PRESENT: YES
 DESCRIPTION:   This is the top level menu for kermit functions.  It gives access
to the send, receive, and edit options.
ITEM: XT-KERMIT RECEIVE                     SYNONYM: R
ITEM: XT-KERMIT SEND                        SYNONYM: S
ITEM: XT-KERMIT EDIT                        SYNONYM: E
  EXIT ACTION: D CLEAN^XTKERM4              ENTRY ACTION: D INIT^XTKERM4
  TIMESTAMP: 61180,30558                    TIMESTAMP OF PRIMARY MENU: 53899,50477
UPPERCASE MENU TEXT: KERMIT MENU

Which OPTIONS item to display:
```

8.2.4 Option Access by User Option

Figure 95: Option Access by User Option



```
Menu Management ...                                                     [XUMAINT]
  Show Users with Selected Primary Menu                               [XUXREF-2]
  Option Access By User                                               [XUOPTWHO]
```

Use the **Show Users with Selected Primary Menu** [XUXREF-2] menu to show which users have been assigned a particular option as a primary or secondary menu. The **Option Access by User** [XUOPTWHO] option is another cross-referencing tool.

8.3 Managing Menus and Options

8.3.1 Managing Primary Menus

When system administrators receive new software applications, existing primary menus should be modified to include the new menus. It is *not* wise to create a new primary menu for every new or unusual circumstance. This would lead to a tremendous variety of menus that would be difficult to sort out and use in the future. Primary menus can be customized with security keys.

REF: For more information on security keys, see the "Security Keys" section.

If there are a few menu options that require special privilege, they can be locked and the security keys assigned to the appropriate users. In this way, a smaller number of primary menus can serve the needs of a larger number of users.

Also, while putting new master menus onto users' secondary menus can be a quick fix, it is *not* a good idea to do this. Too many options on a user's secondary menu can be cumbersome for the user. In addition, in the long run, it is easier for system administrators to manage access to a menu reached from a few well-defined primary menus than to manage access to a menu reached from a large number of users' secondary menus.

8.3.2 Assigning Secondary Menus

An easy way to allocate menu options is to assign them to users individually as SECONDARY MENU OPTIONS (#203) Multiple field entries. Secondary options are unique for each user and are stored in a multiple in the user's NEW PERSON (#200) file entry. Assignment of SECONDARY MENU OPTIONS should be limited to the essential few and should *not* involve deep structures with multiple levels. Instead, new primary menus should be built or existing ones modified. During menu jumping, all branches of both the primary and secondary menu trees are searched each time a jump request is received by the menu system. Greater efficiency and user convenience results if the depth of the secondary menu trees is confined.

8.3.3 ALWAYS SHOW SECONDARIES Field

You can set the ALWAYS SHOW SECONDARIES (#200.11) field in a user's NEW PERSON (#200) file entry. If set to **YES** for a user, that user always has their secondary and common options listed when options on their primary menu are listed, which occurs either by the user entering two question marks (**??**) at the "Select..." menu prompt, or when AUTO MENU is turned on.

8.3.4 Redefining the Common Menu

All users automatically have access to the options on the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu) by virtue of the menu system's design. As described earlier, entering two question marks (**??**) at any select prompt displays the **Common** menu. The only way to deny access to a particular user is to lock the **Common** menu option with a reverse key and then allocate the security key to the same user.

REF: For more information on security keys, see the "Security Keys" section.

The items on the **Common** menu can be left as they are distributed by Kernel, or modified locally as desired. For example, an item can be added to display online help about local computer access policies. This is accomplished by using **Edit options** to edit the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu option. The Item Multiple lists the existing menu choices; other locally namespaced options can be added.

If options are locally added to the standard **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu set, new installations of Kernel do *not* overwrite the changes. During installation, items on the local **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu are compared with the exported items. Any previously exported items that were removed by the site are *not* added back. Brand new items, however, are added and any matching items are updated. Other items that the site may have added are left in place.

8.3.5 Altering Exported Menus

Generally speaking, exported menu structures should stay intact. If local modifications to exported menus are made, great care *must* be taken to preserve any logic that may exist in the exported structure. For example, the entry action of one option can set up key variables that are then assumed to exist when another option, one further down on the menu tree, is invoked. Although each one of a software's options should be able to be invoked independently once the steps described in the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual* for creating and **KILL**ing software-wide variables have been taken (according to the Programming Standards and Conventions [SAC]), this is *not* always the case and *cannot* be assumed.

If an option *cannot* be invoked independently, the developer can set that option's INDEPENDENTLY INVOCABLE field to **NO**, as an alert that some other option or action *must* be done before the option can be called.

To give users the options associated with new software applications, system administrators should try to allocate the menus as whole entities. If dissection appears necessary, the "Internal Relations" section of the software documentation should be consulted before rearranging any of the items.

8.3.6 Delete Unreferenced Options Option

Figure 96: Delete Unreferenced Options Option

```
Programmer Options ...  <locked:  XUPROG>                             [XUPROG]
  Delete Unreferenced Options                               [XQ UNREF’D OPTIONS]
```

All options for interactive use (*not* designed exclusively as queueable tasks) should normally be tied to a menu that is used as a primary menu or at least as a secondary menu. Standalone options that have no parents and are *not* menu-type options should be reviewed. They may be obsolete software options or local test options and could be candidates for deletion. Use the **Delete Unreferenced Options** [XQ UNREF'D OPTIONS] option to delete unreferenced options. It can be used to cycle through the entire OPTION (#19) file and delete *non*-menu options that are *not* referenced by other options. Deletion should obviously be done with care. Use of this option is limited to those who hold the XUPROG security key.

8.3.7 Fix Option File Pointers Option

Figure 97: Fix Option File Pointers Option

```
Menu Management ...                                                     [XUMAINT]
  Fix Option File Pointers                                             [XQOPTFIX]
```

After performing maintenance work on the OPTION (#19) file (e.g., deleting obsolete options that may have been items on a menu), you can use the **Fix Option File Pointers** [XQOPTFIX] option (see Figure 98) to remove any dangling pointers that may have been left in the Item multiple. Running this option is an alternative to having VA FileMan update the pointers each time an individual option is deleted.

Figure 98: Fix Option File Pointers Option—Sample User Dialog

```
Select OPTION NAME: ZZTEST3 <Enter>          Test Option
NAME: ZZTEST3// @
   SURE YOU WANT TO DELETE THE ENTIRE ‘ZZTEST3’ OPTION? Y <Enter> (YES)
SINCE THE DELETED ENTRY MAY HAVE BEEN ‘POINTED TO’
BY ENTRIES IN THE ‘USER’ FILE, ETC.,
DO YOU WANT THOSE POINTERS UPDATED (WHICH COULD TAKE QUITE A WHILE)? NO// <Enter>
```

8.3.8 Testing a User’s Menus

Figure 99: Switch Identities Option

```
User Management...                                                        [XUSER]
  Switch Identities                                                  [XUTESTUSER]
```

You can test a user’s menus using the **Switch Identities** [XUTESTUSER] option. It lets you test the user’s menus and security keys. It does *not* allow you to execute any bottom-level menu options, however; it only lets you navigate menu trees. You are reminded at each prompt whose menu it is that you are testing. To exit this mode and return to your own menus, simply enter an asterisk (*).

8.3.9 Managing Out-Of-Order Option Sets

Figure 100: Out-Of-Order Set Management Menu Options

```
Menu Management ...                                                    [XUMAINT]
  Out-Of-Order Set Management...                                      [XQOOMAIN]
    Create a Set of Options To Mark Out-Of-Order                      [XQOOMAKE]
    List Defined Option Sets                                          [XQOOSHOW]
    Mark Option Set Out-Of-Order                                       [XQOOFF]
    Options in the Option File that are Out-of-Order                [XQOOSHOFIL]
    Protocols Marked Out-of-Order in Protocol File                  [XQOOSHOPRO]
    Recover Deleted Option Set                                        [XQOOREDO]
    Remove Out-Of-Order Messages from a Set of Options                  [XQOON]
    Toggle options/protocols on and off                                [XQOOTOG]
```

Menu Manager, starting with Kernel 8.0, provides a mechanism for defining sets of options and protocols, and a way to disable and enable access for these pre-defined option and protocol sets via options on the **Out-Of-Order Set Management** [XQOOMAIN] menu. This can be handy when you need to repeatedly disable and enable sets of options and protocols.

Use the **Create a Set of Options to Mark Out-Of-Order** [XQOOMAKE] option to define a set of options. You are prompted first to select options, and then to select protocols.

For both options and protocols, you can use the following to:

- Add a group of options to the set—Use the wildcard asterisk (*****) with or without a namespace.
- Add a range of options to a set—Use **NAM1-NAM2** to add a range of options from **NAM1** to **NAM2** to the set, where “**NAM**” represents a namespace.
- Subtract/Remove a group of options from a set—Use the minus sign (i.e., hyphen, **-**) followed by a namespace.

Use the **Mark Option Set Out-Of-Order** [XQOOFF] option to disable access to a set of options. You are asked to enter the message used to place all options in the set out-of-order. The option then places the message in each option’s OUT OF ORDER MESSAGE (#2) field.

Use the **Remove Out-Of-Order Messages from a Set of Options** [XQOON] option to enable access to an option set.

To toggle the status of an individual option only, use the **Toggle Options/Protocols On and Off**[XQOOTOG] option.

Out-of-Order Option sets are stored in the **^XTMP** global, with a purge date set for **seven days** in the future. If you place a set of options out of order, but the option set is purged from **^XTMP** before you enable access to it, you can rebuild the out-of-order option set using the **Recover Deleted Option Set** [XQOOREDO] option. It asks you to specify the exact text of the message used to place the set of options out of order; it then recreates an out-of-order option set containing all options currently placed out of order with the specified message

NOTE: Make sure the message you specify is unique to the set of options you are re-enabling.

You can then enable access to the rebuilt option set with the **Remove Out-Of-Order Messages from a Set of Options** [XQOON] option.

To see what sets of options have been grouped in sets on the system, use the **List the Defined Options Sets** [XQOOSHOW] option. To show all options and protocols currently marked out of order, use the **Options in the Option File that are Out-of-Order** [XQOOSHOFIL] option and the **Protocols Marked Out-of-Order in Protocol File** [XQOOSHOPRO] option.

8.4 Restricting Option Usage

Figure 101: Restrict Availability of Options Option

```
Menu Management ...                                                [XUMAINT]
  Restrict Availability of Options                               [XQRESTRICT]
```

Options can be restricted in terms of when users can select them and when devices can be used to invoke them. Many of the option restrictions are included in the **Restrict Availability of Options** [XQRESTRICT] option.

8.4.1 Setting Options Out of Order

To completely restrict access, you can mark an option to be out-of-order. Do this by entering text in an option's OUT OF ORDER MESSAGE (#2) field in the OPTION (#19) file. If a user attempts to invoke the option, the Out of Order Message is displayed.

8.4.2 Locks

Both the normal lock and Reverse/Negative lock can be associated with options (as described in the "Security Keys" section). Also, M code can be entered in the HEADER, ENTRY ACTION, or EXIT ACTION fields to restrict the use of an option given certain conditions.

8.4.3 Prohibited Times

You can prohibit the use of an option at certain times during the day by assigning a set of prohibited time periods at the "Select TIMES PROHIBITED" prompt. Options scheduled to run through TaskMan will also be prohibited from running during these prohibited times.

8.4.4 Permitted Devices

If the RESTRICT DEVICES flag is set to **YES**, the option can only be invoked on one of the devices listed in the PERMITTED DEVICES Multiple field. Thus, the running of an option can be restricted. This flag does *not* affect the choice of devices used for the output from options. It instead controls the processing involved in the use of the option itself.

8.4.5 QUEUING REQUIRED Flag

Using the **Edit options** [XUEDITOPT] option, you can allow users to invoke an option, but force any output to be queued outside of certain times of day, by editing the option's QUEUING REQUIRED Multiple field. In this multiple's TIME PERIOD (#.01) and DAY(S) FOR TIME PERIOD (#.02) fields enter the time periods and days in which you do *not* want the option's output to be produced. During these time periods, the output of the options can only be queued. When a user requests a time for queuing, the menu system determines the next permissible day and time for output. Thus, users can invoke the option and use it to define the parameters for the subsequent processing, but the actual work is done during a later time period, presumably when the system is less busy.

8.5 Menu Manager Options that Should Be Scheduled

This section describes the two Menu Manager options that should be regularly scheduled.

Kernel exports a number of other options that should be scheduled to run at regular intervals. Most of these are located on the Parent of Queuable Options [ZTMQUEUABLE OPTIONS] menu.

REF: For a complete list, along with suggested scheduling frequencies, see the *Kernel Installation Guide*.

8.5.1 Clean Old Job Nodes in XUTL Option

The **Clean old Job Nodes in the XUTL** [XQ XUTL $J NODES] option is Kernel's purge option for Kernel globals. This option purges the following globals:

- **^XUTL**
- **^UTILITY**
- **^TMP**
- **^XTMP**
- **^XUSEC**

Figure 102: Clean old Job Nodes in XUTL Option

```
Operations Management ...                                        [XUSITEMGR]
    Clean old Job Nodes in XUTL                              [XQ XUTL $J NODES]
```

User stacks for each user's job are stored in the **^XUTL** global.

REF: For more information, see the "^XUTL Global: Structure and Function" section.

This is also called the compiled menu system. If a job ends abnormally (e.g., upon error, UCI switching, or developer exits that bypass **^XUS**), the entries remain in the global (this explains why developers are advised to halt out of programmer mode with **D ^XUSCLEAN** rather than simply halting.)

The purge routine sets a purge date of **seven days** in the past. Any user stack in **^XUTL** older than seven days is purged. Any entries with a matching **$J** at the top level of **^UTILITY** and **^TMP** are also **KILL**ed.

Next, after cleaning out the user stacks in **^XUTL**, the purge routine checks **^UTILITY** and **^TMP**. Any entry at subscript (**$J**) or (**namespace, $J**) that does *not* have a matching entry in the user stacks in **^XUTL** is **KILL**ed.

Next, the purge routine checks **^XTMP**. Any entry in **^XTMP** at subscript (namespace) lacking a header node at (**namespace,0**), or with a purge date in the header node less than the purge date determined by the purge routine is **KILL**ed.

Finally, the purge routine goes through the signon nodes stored at **^XUSEC(0,“CUR”,DUZ,DATE)**. Any nodes older than the purge date are **KILL**ed.

The XQ XUTL $J NODES option should be queued to run on a regular basis. If separate copies of the **^XUTL** global are maintained on different CPUs, separate entries should be made in the OPTION SCHEDULING (#19.2) file for each CPU, so that a separate job purges each CPU’s **XUTL** global. Because this option deletes any user stacks that are time-stamped with a date earlier than the purge date determined by this option (seven days) you need to take care how frequently you schedule it (in the unusual event of a seven-day long job, this option should obviously *not* be run).

8.5.2 Rebuilding Primary Menu Trees

Figure 103: Building Primary Menu Trees Options

```
PARENT OF QUEUABLE OPTIONS                                    [ZTMQUEUABLE OPTIONS]
  Non-interactive Build Primary Menu Trees                         [XQBUILDTREEQUE]
Menu Management ...                                                       [XUMAINT]
  Build Primary Menu Trees                                            [XQBUILDTREE]
```

The menu system uses local menu trees to process requests. When changes are made to the menu structure, the local menu trees are rebuilt (a process also known as microsurgery). If a user attempts an “Up-arrow Jump” when the local trees need to be rebuilt or are being rebuilt, a message is issued about quick access being temporarily disabled; the user is *not* able to jump to reach the option. Microsurgery is triggered in the following situations:

- The **Edit options** [XUEDITOPT] option is used.
- An Out-of-Order option set is enabled or disabled.
- A sufficiently large number of changes have been made to a menu tree.

It is also *recommended* to rebuild all primary menu trees every other day during non-peak hours, using the XQBUILDTREEQUE option. If separate copies of **^XUTL** are maintained on different CPUs, separate entries should be made in the OPTION SCHEDULING (#19.2) file for each CPU so that a separate job rebuilds each CPU's **^XUTL** global.

Primary menu trees can also be built/repaired immediately using the **Build Primary Menu Trees** [XQBUILDTREE] option. In particular, if menu jumping has stopped working and microsurgery is *not* fixing the menus, use the **Build Primary Menu Trees** [XQBUILDTREE] option to force a menu rebuild to fix the problem.

8.6 Error Messages during Menu Jumping

There are some conditions under which a menu jump may *not* be completed. In these cases the user sees one of the following error messages (Figure 104 to Figure 109):

Figure 104: Menu Jump Error Message (1 of 6)

```
I NEED TO REBUILD MENUS .... QUICK ACCESS IS TEMPORARILY DISABLED  Please proceed
to {target option's menu text}
```

This means that the time stamps on the OPTION (#19) file and the **^XUTL** global indicate that the OPTION (#19) file has been modified since the menus were compiled in **^XUTL** and the global is therefore locked until **XQ8** can recompile the modified menus. This error message can be generated by both user-generated jumps and phantom jumps.

Figure 105: Menu Jump Error Message (2 of 6)

```
*** WARNING ***
Illegal jump requested to option '{option's menu text}'  Jump pathway locked at
option '{locked option's menu text}'
```

This indicates that a locked option for which the user does *not* possess the security key has been encountered in the tree between the option where the jump was requested and the target option to which the jump was requested. This error message can be generated by both user-generated jumps and phantom jumps.

Figure 106: Menu Jump Error Message (3 of 6)

```
*** WARNING ***
Illegal jump was requested to option '{option menu text}'  Jump path out of order
from '{option's menu text}'  with message '{out of order message}'
```

This means that an option on the tree between the option where the phantom jump was requested and the target option has been marked as out of order (OUT OF ORDER MESSAGE [#2] Field of the OPTION [#19] file). This error message can be generated by both user-generated jumps and phantom jumps.

Figure 107: Menu Jump Error Message (4 of 6)

```
*** WARNING ***
Illegal jump was requested to option '{option menu text}'  Variable XQUIT
encountered at option '{option name}'
```

This means that the jump logic has encountered the variable **XQUIT** (detected with a **$DATA** statement). This variable is usually set by an Entry Action (Field #20 of the OPTION [#19] file) and causes the menu system to refuse to run or jump past that option. This error message can be generated by both user-generated jumps and phantom jumps.

Figure 108: Menu Jump Error Message (5 of 6)

```
*** WARNING ***
Background jump requested to option '{value in XQMM("J")}'  but this option does
not exist on your system.
```

A VA FileMan lookup was attempted for the option set in the variable **XQMM("J")** but no such option was found in the OPTION (#19) file. This error message can only be generated from a phantom jump.

Figure 109: Menu Jump Error Message (6 of 6)

```
*** WARNING ***
Background jump requested to option '{option's menu text}' but  you do not have
access to this option.  See your computer  representative.
```

This means that the target option requested by **XQMM("J")** is *not* in the tree of options to which this user has access (that is, the target option was neither in the user's primary menu tree nor specifically listed as a secondary menu for that user). This error message can only be generated from a phantom jump.

REF: For more information on phantom jumps, see the "Menu Manager: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

8.7 ^XUTL Global: Structure and Function

The **^XUTL** global is an account-specific global. It should exist in each Production account on your system. This global is created primarily from information in the OPTION file [**^DIC(19)**] and is therefore sometimes referred to as "the compiled menu system."

^XUTL is divided into three main sections:

- **User Stacks**

  ^XUTL("XQ",$J)

  ^XUTL("XQT",$J) (MENU templates only)

- **Display Nodes**

  ^XUTL("XQO",ien)

- **Jump Nodes**

  ^XUTL("XQO","P"_ien)

8.7.1 User Stacks

User stacks are stored in nodes in **^XUTL("XQ",$J)** and **^XUTL("XQT",$J)**.

The example illustrated in Figure 111 shows a typical user stack. In this case the **$J** is **541065826**.

The "**XQ**" nodes can be divided into meaningful sets according to what is contained in the third subscript. The numeric third subscripts begin with the **zero** node, which is set to the date and time in VA FileMan format by the program **^XUS1** when the user logs on or **^%XUCI** when the user is changing UCIs.

The other numeric, third subscripts (in this case the numbers **1** to **3**) reflect the user's progression through the menu system.

Each time a new option is invoked, a new node is created that contains the following:

- Option number, concatenated with a **P.**
- Number of the option whose compiled menu tree contains the current option.
- A caret (^).
- **Zero**-node of the OPTION (#19) file for that option.

A different format is used for options in a user's secondary menu tree.

A pointer in the node **^XUTL("XQ", $J, "T")** indicates which option in this list of numbered nodes the menu driver is currently using. This pointer is set and reset by the menu driver as the user moves up and down the menu tree. In the example, XUPROGMODE is the option that the menu driver is currently using.

Other “**XQ**” nodes of the global that have a non-numeric third subscript are used to store various pieces of Kernel information that are set up at signon. **^XUTL(“XQ”,$J,“XQM”)** points to the user’s primary menu.

In the example in Figure 110, the user’s primary menu is OPTION (#19) file entry #29.

Figure 110: User Stack Example

```
^XUTL(“XQ”,541065826,0) = 2920113.081624
^XUTL(“XQ”,541065826,1) = 29P29^EVE^Systems Manager
                          Menu^^M^.5^^192^^^^^^n^1^^^
^XUTL(“XQ”,541065826,2) = 31P29^XUPROG^Programmer Options^^M^^
                          XUPROG^^^^^^^n^^
^XUTL(“XQ”,541065826,3) = 49P29^XUPROGMODE^Programmer mode^^R
                          ^^XUPROGMODE^^^^^^^ n^^
^XUTL(“XQ”,541065826,”DUZ”) = 63
^XUTL(“XQ”,541065826,“DUZ(0)”) = LlPp
^XUTL(“XQ”,541065826,“DUZ(2)”) = 16000
^XUTL(“XQ”,541065826,“IO”) = _TNA5103:
^XUTL(“XQ”,541065826,“IOBS”) = $C(8)
^XUTL(“XQ”,541065826,“IOF”) = #,$C(27,91,50,74,27,91,72)
^XUTL(“XQ”,541065826,“ION”) = LAT DEVICE
^XUTL(“XQ”,541065826,“IOS”) = 158
^XUTL(“XQ”,541065826,“IOSL”) = 24
^XUTL(“XQ”,541065826,“IOST”) = C-VT100HIGH
^XUTL(“XQ”,541065826,“IOST(0)”) = 149
^XUTL(“XQ”,541065826,“IOT”) = VTRM
^XUTL(“XQ”,541065826,“IOXY”) = W $C(27,91)_((DY+1))_$C(59)_((DX+1))_$C(72)
^XUTL(“XQ”,541065826,“T”) = 3
^XUTL(“XQ”,541065826,“XQM”) = 29
```

8.7.2 XQT Nodes (MENU Templates)

The “**XQT**” nodes are used to create a stack of options similar to the “**XQ**” stack when a MENU template is invoked. These nodes are translated from the **^VA(200,DUZ,19.8)** Multiple when a user precedes an option selection with a left square bracket character, “**[**”, much like a PRINT template is invoked in VA FileMan. For example, if the user has defined a MENU template named “**DOIT**” using the Menu Template options of the User’s Tool Box, typing “**[DOIT**” loads that sequence of options into the “**XQT**” nodes and begins executing them. When a MENU template is requested by the user, the option tree of that template is loaded into the “**XQT**” nodes and remains loaded as long as the user is logged on. Further requests for “**[DOIT**” uses that same stack.

8.7.3 Display Nodes

Display nodes are stored in **^XUTL(“XQO”,** internal number).

The first example in Figure 112 shows the display nodes for **EVE**, the System Manager’s Menu. The internal number of **EVE** in this particular OPTION (#19) file is **29**. In the first part of the example the option names and menu texts, along with a limited number of fields for that option compiled from the OPTION (#19) file, are concatenated together. It is from this part that **XQ2** (the menu display program) gets the information it needs.

In the second part, all the menu texts and synonyms are listed in order in uppercase. It is here that **XQ** tries to match what the user entered at the terminal with the correct option. The third part of the example, the **0**th node of the options, is listed by number and provides the remaining information that the Menu System may need to make the option work. To understand what the various ^ pieces mean, look at a VA FileMan global format data dictionary listing of the OPTION (#19) file.

Illustrated in the second example in Figure 112 is the display node for the SECONDARY MENU OPTIONS of a user whose **DUZ** is equal to **66**. Here, the user has only a single secondary menu called “Secondary Menu” (with an internal number of **580** in the OPTION [#19] file). The various parts of this example are identical to those of the Display Nodes for the **EVE** menu example (Figure 111).

> **NOTE:** The second subscript, instead of pointing to a menu in the OPTION (#19) file, is a “U” concatenated with the user’s **DUZ** which points to the NEW PERSON (#200) file entry. This is because secondary menu options are stored in the SECONDARY MENU OPTIONS (#203) Multiple field in the NEW PERSON (#200) file entry for each user.

Figure 111: Display Nodes for EVE Example

```
^XUTL(“XQO”,29,0) = 2^55048,38923
^XUTL(“XQO”,29,0,1) = ^XUCORE^Core Applications ...^NOT
                         AVAILABLE^^^^^^XUTIO^Device Handler
                         ...^^^^n^^FM^DIUSER^VA FileMan ...^^^^n^^^XMMGR^
                         Manage Mailman ...^^^^^^^XUMAINT^Menu Management
                         ...^^^^n^^^XUPROG^Programmer Options ...^^XUPROG^^^
                           ...^
^XUTL(“XQO”,29,0,2) = ^XUSITEMGR^Operations Management ...^^^^^^^XU-SPL-MGR
                           ^Spool Management ...^^^^^^^XUSPY^System Security
                         ...^^^^^^^ZTMMGR^Task Manager ...^^^^n^^^XUSER^User
                         Edit ...^^^^^^
^XUTL(“XQO”,29,“CORE APPLICATIONS”) = 40^1
^XUTL(“XQO”,29,“DEVICE HANDLER”) = 32^1
^XUTL(“XQO”,29,“FM”) = 19^0
^XUTL(“XQO”,29,“MANAGE MAILMAN”) = 30^1
^XUTL(“XQO”,29,“MENU MANAGEMENT”) = 9^1
^XUTL(“XQO”,29,“OPERATIONS MANAGEMENT”) = 174^1
^XUTL(“XQO”,29,“PROGRAMMER OPTIONS”) = 31^1
^XUTL(“XQO”,29,“SPOOL MANAGEMENT”) = 415^1
^XUTL(“XQO”,29,“SYSTEM SECURITY”) = 226^1
^XUTL(“XQO”,29,“TASK MANAGER”) = 83^1
^XUTL(“XQO”,29,“USER EDIT”) = 39^1
^XUTL(“XQO”,29,“VA FILEMAN”) = 19^1
^XUTL(“XQO”,29,“^”,9) = ^XUMAINT^Menu Management^^M^^^105^^^n^n^^n^^^^
^XUTL(“XQO”,29,“^”,19) = FM^DIUSER^VA FileMan^^M^^^^^^n^^^n^1^^
^XUTL(“XQO”,29,“^”,30) = ^XMMGR^Manage Mailman^^M^^^299^^^^^54^^1^1^^^
^XUTL(“XQO”,29,“”^”,31) = ^XUPROG^Programmer Options^^M^^XUPROG^^^^^^^n^^
^XUTL(“XQO”,29,“^”,32) = ^XUTIO^Device Handler^^M^^^413^^^n^^20^n^^
^XUTL(“XQO”,29,“^”,39) = ^XUSER^User Edit^^M^^^153^^^^^^n^^
^XUTL(“XQO”,29,“^”,40) = ^XUCORE^Core Applications^1^M^^^^^^^^^n^^
^XUTL(“XQO”,29,“^”,83) = ^ZTMMGR^Task Manager^^M^^^^^^n^^50^^1^^
^XUTL(“XQO”,29,“^”,174) = ^XUSITEMGR^Operations Management^^M^^^^^^^y^^n^^
^XUTL(“XQO”,29,“^”,226) = ^XUSPY^System Security^^M^^^^^^^^119^n^^
^XUTL(“XQO”,29,“^”,415) = ^XU-SPL-MGR^Spool Management^^M^^^419^^^^^20^^
```

Figure 112: Display Nodes for a Secondary Menu

```
^XUTL(“XQO”,“U66”,0) = 1^54927,30758
^XUTL(“XQO”,“U66”,0,1) = ^ZZTSTSM^Secondary Menu ...^^^^n^^
^XUTL(“XQO”,“U66”,“SECONDARY  MENU”) = 580^1
^XUTL(“XQO”,“U66”,“^”,580) = ^ZZTSTSM^Secondary Menu^^M^^^^^^n^^^^1^1^^1
```

8.7.4 Jump Nodes

Jump nodes are stored in **^XUTL(“XQO”,“P”_internal number)**, where there is one “**P_...**” entry in **^XUTL(“XQO”)** for each primary menu that exists. The jump nodes, for each primary menu, store the pathways to all options that can be jumped to.

The jump nodes are created in the **XQ8*** series of programs. They are very similar to display nodes, except that:

- They have a **P** concatenated on the front of the primary option’s number in the second subscript.
- These nodes describe the entire primary menu tree rather than just the single level tree.

Examples of the jump nodes for a single primary menu are shown in Figure 113 and Figure 114. Since these nodes can be very extensive in number, some nodes have been removed from the examples to save space.

In the first example (Figure 113) are the “lookup” nodes, where the jump software tries to match a menu text or synonym with what the user has entered at the terminal. Each node is set to its internal number in the OPTION (#19) file and, in the second ^ piece, a:

- **0**—If it is a synonym.
- **1**—If it is menu text.

In the second example (Figure 114), the “menu pathway” entries below the “**P580**” node show all of the options that can be jumped to from the primary menu whose internal entry number (IEN) is **580**. Each entry contains lists of the series of options that *must* be navigated through in a jump from the primary menu. In the case of the option DILIST (# 17), the list of options that have to be processed is **520,519,518,411,17**. If, as in the case of ZZTEST4 (# 318), there is more than one possible pathway, then each is listed along with various other necessary pieces of information (e.g., locks, time restraint, etc.).

Figure 113: Jump Nodes Example—Lookup Nodes

```
^XUTL(“XQO”,“P580”,0) = 55165,28536
^XUTL(“XQO”,“P580”,“19^”) = 394^0
^XUTL(“XQO”,“P580”,“2ND SECOND LEVEL MENU TEST^”) = 575^1
^XUTL(“XQO”,“P580”,“3^”) = 518^0
^XUTL(“XQO”,“P580”,“ACTN^”) = 391^0
^XUTL(“XQO”,“P580”,“ALL^”) = 420^0
```

Figure 114: Jump Nodes Example—Menu Pathways

```
^XUTL(“XQO”,“P580”,“LIST FILE ATTRIBUTES^”) = 17^1
^XUTL(“XQO”,“P580”,“TEST 4^”) = 318^1
...
^XUTL(“XQO”,“P580”,“TOOL^”) = 581^0
^XUTL(“XQO”,“P580”,“X-TYPE OPTION TEST^”) = 576^1
^XUTL(“XQO”,“P580”,“X^”) = 576^0
^XUTL(“XQO”,“P580”,“ZDAVE^”) = 411^1
^XUTL(“XQO”,“P580”,“^”,5) = ^XUEDITOPT^Edit
                               options^^E^581,5,^^106^^^^^20^n^^^^
^XUTL(“XQO”,“P580”,“^”,17) = ^DILIST^List File Attributes^^A^
                               520,519,518,411,17,^^^^^n,^y^^n^1^^^
...
^XUTL(“XQO”,“P580”,“^”,318) = ^ZZTEST4^Test
                               4^^O^520,575,397,318,^^^^^n,^^^^^^
^XUTL(“XQO”,“P580”,“^”,318,0) = 2
^XUTL(“XQO”,“P580”,“^”,318,0,1) = 520,575,578,397,318,^^^n,^
^XUTL(“XQO”,“P580”,“^”,318,0,2) = 520,575,578,318,^^^n,^
...
^XUTL(“XQO”,“P580”,“^”,579) = ^ZZLEVEL3B^Phantom
                               Mother^^M^520,575,579,^^^^^n,^^^^1^1^^1
^XUTL(“XQO”,“P580”,“^”,580) = ^ZZTSTPM^Primary Menu^^M^^^^^^n^^^^1^1^^1
^XUTL(“XQO”,“P580”,“^”,581) = ^ZZLUKTOOLS^Luke’s
                               Tools^^M^581,^^^^^^^^^1^1^^1
```

8.8 Menu Startup Parameter

The XQ MENUMANAGER PROMPT parameter is checked during menu startup. It allows sites to change the default **<TEST ACCOUNT>** prompt to another value (e.g., **<LEGACY SYSTEM>**) in menu prompts of *non*-Production VistA systems. The text defined by this parameter is inserted in the Menu Manager prompts. If no text is defined, the hard-coded default is “ **<TEST ACCOUNT>**”. Alternatives could be:

- **“ <LEGACY SYSTEM>**
- “ **<CONTINGENCY>**”
- “ **<READ ONLY>**”
- Any other value from 3 to 20 characters, depending upon the purpose of the *non*-Production VistA system.

To change the value on a *non*-Production system, use the **General Parameter Tools** [XPAR MENU TOOLS] option and select “**EP Edit Parameter Values**.” You have to log off and log back into VistA to see the changed menu prompt.

NOTE: The prompt can be set in advance on a Production system before it is mirrored to a *non*-Production system, and the prompt only appears on the *non*-Production system.

NOTE: The XQ MENUMANAGER PROMPT parameter was released with Kernel Patch XU*8.0*614.

8.9 Menu Manager Variables (Troubleshooting)

Table 12 lists the Menu Manager variables that are always defined. It may be useful for system administrators to know what these variables signify when investigating errors. If an error is reported in VA FileMan's **DIP** routine, for example, knowing the value of **XQY** at the time of the error indicates which option was invoking the **DIP** routine. The option can then be reviewed to discover the name of the routine that was calling **DIP**.

Table 12: Menu Manager Variables (Always Defined)

Variable	Description
XQABTST	Flag that signals whether alpha-beta testing is in effect.
XQDIC	Internal entry number (IEN) of the option's parent (which *must* be a menu) in the OPTION (#19) file, if an option is executing. If the user is in a menu, **XQDIC** is set to the IEN of the current menu's parent (unless they are in their primary menu, in which case **XQDIC** is set to the IEN of the primary menu). The value of **XQDIC** also corresponds to the second subscript in the display nodes portion of the **^XUTL** global, **^XUTL("XQO",)** for the menu in question.
XQPSM	Like **XQDIC**, a lookup value into the second subscript of **^XUTL**, the compiled menu global. **XQPSM** points to the tree of the target option in the jump. It resulted from the ability to jump to any option, *not* just ones on the primary menu tree. It can help identify jumps from a primary, secondary, or Common option.
XQT	Current option's type (e.g., **M** for menu, **A** for action).
XQUR	User's response to the menu prompt (replaces **A**).
XQUSER	User's name in the form SEVEN A. XUUSER.
XQY	Internal entry number (IEN) of the current option or menu (replaces **Y**).
XQY0	First node (subscript of **zero**) of the current option [replaces **Y(0)**].
XQXFLG	Contains several flags, including whether capacity management testing is active.

8.10 Security Keys

8.11 User Interface

Security keys are primarily used to allow access to specially protected options. If a software application exports a menu that has one or two options that require a secured level of access, they can use security keys to lock those special options. When an option is locked, you can only use the locked option if you hold the security key matching the key with which the option was locked.

Entering two question marks (**??**) at the menu system's select prompt displays the current options. If any of the options are locked, that fact is listed also, along with the names of any associated security keys. In the example in Figure 115, the option Programmer Options is locked with a security key named XUPROG:

Figure 115: Sample Locked Menu Options Showing Required Security Key—Entering Two Question Marks (??)

```
Select Systems Manager Menu Option: ??
          Device Handler ...                                            [XUTIO]
          Menu Management ...                                         [XUMAINT]
          Programmer Options ...                                       [XUPROG]
             **> Locked with XUPROG
```

You can list which security keys you currently hold by using the **Display User Characteristics** [XUUSERDISP] option on the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). It displays a list of all security keys you hold, similar to Figure 116:

Figure 116: Display User Characteristics Option—Sample Output

```
KEYS HELD
---------
   XUPROG          XUMGR          XUPROGMODE     XUAUTHOR       ZTMQ
```

The security keys you need to carry out computing activities should be assigned by system administrators when your computer account is first added to the system. Other keys can be allocated at a later time by system administrators or designee (e.g., an application coordinator) with the use of the Secure Menu Delegation menu utilities.

8.12 System Management

8.12.1 Identifying Locked Options

System administrators can list which security keys lock what options by using the **Diagram Menus** [XUUSERACC] option, which is located on the **Display Menus and Options** [XQDISPLAY OPTIONS] menu under the Menu Management [XUMAINT] menu. Figure 117 shows that the **Programmer Options** [XUPROG] menu is locked with the XUPROG security key. It also shows that one of its options, **Programmer mode** [XUPROGMODE], is locked with the XUPROGMODE security key:

Figure 117: Diagram Menus Option—Sample User Dialog

```
Select Menu Management Option: DIAGRAM MENUS
Select USER (U.xxxxx) or OPTION (O.xxxxx) name: O.XUPROG
Programmer Options (XUPROG)
**LOCKED: XUPROG**
--------------------------PG Programmer mode
                 [XUPROGMODE]
                   **LOCKED: XUPROGMODE**
```

Security keys are stored in the SECURITY KEY (#19.1) file. Security keys given to users are stored in the users’ NEW PERSON (#200) file entries, in the KEYS Multiple field.

Options are locked by a given security key when the name of that key is entered into the LOCK (#3) field of the OPTION (#19) file. If an option is locked, users need to be given the security key in order to invoke the option.

8.12.2 Key Management

Keys are defined and allocated to users with options on the **Key Management** [XUKEYMGMT] menu, as shown in Figure 118:

Figure 118: Key Management Menu Options

```
SYSTEMS MANAGER MENU ...                                                      [EVE]
  Menu Management ...                                                     [XUMAINT]
     Key Management ...                                                 [XUKEYMGMT]
       Allocation of Security Keys                                       [XUKEYALL]
       De-allocation of Security Keys                                  [XUKEYDEALL]
       Enter/Edit of Security Keys                                      [XUKEYEDIT]
       All the keys a user needs                                          [XQLOCK1]
       Change user’s allocated keys to delegated keys                [XQKEYALTODEL]
       Keys for a given menu tree                                         [XQLOCK2]
       Delegate keys                                                     [XQKEYDEL]
       List users holding a certain key                                  [XQSHOKEY]
       Remove delegated keys                                            [XQKEYRDEL]
       Show the keys of a particular user                               [XQLISTKEY]
```

8.12.3 Allocating and De-allocating Security Keys

The main option to assign security keys to a user or users is the **Allocation of Security Keys** [XUKEYALL] option. Allocating a security key to a user lets them invoke options that are locked with the key. For options with reverse locks, allocating the security key locks the user out from the option. In either case, allocating the key to a user does *not* allow the user to give the key to anyone else.

REF: For more information on reverse locks, see the "Using Security Keys with Reverse Locks" section.

NOTE: The PSDRPH security key *cannot* be allocated using this option, the **Allocate/De-Allocate of PSDRPH Key (Audited)** [PSO EPCS PSDRPH KEY] option *must* be used to allocate this security key.

To remove a security key from a user, use the **De-allocation of Security Keys** [XUKEYDEALL] option.

Unless you have been delegated a security key, the only way you can allocate or de-allocate keys is if you hold the XUMGR security key or have a FILE MANAGER ACCESS CODE (#3) field of @.

REF: For more information on delegating security keys, see the "Delegating Security Keys" section.

NOTE: The PSDRPH security key *cannot* be de-allocated using this option, the **Allocate/De-Allocate of PSDRPH Key (Audited)** [PSO EPCS PSDRPH KEY] option must be used to de-allocate this security key.

All of the security keys that a new user needs to use their assigned options can be determined by using the **All the Keys a User Needs** [XQLOCK1] option on the **Key Management** [XUKEYMGMT] menu. This produces a list of the primary and secondary menus for that user and compiles a list of the keys for that menu tree. This list can then be assigned or delegated. It can also be edited before the keys are given to the user. Similarly, the **Keys For a Given Menu Tree** [XQLOCK2] option examines a menu and lists all of the security keys associated with all sibling options.

8.12.4 Delegating Security Keys

Delegating keys allows you to give a user the ability to assign specific security keys to other users, as opposed to the XUMGR security key and @ VA FileMan Access code (i.e., FILE MANAGER ACCESS CODE [#3] field), which allow all keys to be assigned.

One way to delegate security keys is to use the **Change user's allocated keys to delegated keys** [XQKEYALTODEL] option. This option delegates to a user all of the security keys that are currently allocated to that user. Any entries in their KEYS Multiple field are entered in the DELEGATED KEYS Multiple field as well. They can now use the **Allocation of Security Keys** [XUKEYALL] option to give the security keys to others.

Alternatively, system administrators can use the **Delegate keys** [XQKEYDEL] option to populate the DELEGATED KEYS Multiple field one-by-one.

A user who has been delegated a security key can allocate that key to others in two ways:

- Through the **Allocation of Security Keys** [XUKEYALL] option, if it is on their menu.
- By delegating an option locked by the security key in question; the key is allocated along with the option.

The key recipients (excepting holders of the XUMGR security key or a FILE MANAGER ACCESS CODE [#3] field of @) *cannot* assign the security key to others, however, even if they have access to the **Allocation of Security Keys** [XUKEYALL] option, because the key does *not* exist in their DELEGATED KEYS Multiple field.

One example of key delegation is a system administrator designee, delegated the Provider key, who allocates that key to incoming medical residents.

For security reasons, users who have a key in their DELEGATED KEYS Multiple field *cannot* allocate that key to themselves. That key *must* be awarded by another user who has been delegated the key or by a system administrator who holds the XUMGR system security key.

8.12.5 Creating and Editing Security Keys

Keys can be created using the **Enter/Edit of Security Keys** [XUKEYEDIT] option on the **Key Management** [XUKEYMGMT] menu. If a security key has already been defined, its name *cannot* be edited. It also *cannot* be deleted, as discussed below. Other key attributes stored in the SECURITY KEY (#19.1) file can be used for special purposes. Attributes of the Provider key are shown in the example in Figure 119:

Figure 119: Attributes for the Provider Security Key—Sample User Dialog

```
Select SECURITY KEY NAME: PROVIDER
```

No editing.

```
NAME: PROVIDER// <Enter>
DESCRIPTIVE NAME: Provider// <Enter>
PERSON LOOKUP: LOOKUP// <Enter>
KEEP AT TERMINATE: YES// <Enter>
DESCRIPTION:
  1>This KEY is given to all entries in the New Person file that need
  2>to be looked up as a Provider. Those entries that hold this key
  3>are considered to be providers.  It was given to all active
  4>Providers in file 6 at the time of the Kernel 7 install.
EDIT Option: <Enter>
Select SUBORDINATE KEY: <Enter>
GRANTING CONDITION: <Enter>
```

8.12.5.1 PERSON LOOKUP

As described in the “Security Keys: Developer Tools” section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer’s Guide*, a special AK cross-reference on the NEW PERSON (#200) file is maintained automatically for anyone who is granted a security key that is flagged for Person Lookup. This cross-reference has been introduced to facilitate identification of user groups, like providers.

8.12.5.2 KEEP AT TERMINATE

As described in the “Signon/Security” section concerning user deactivation, security keys that are marked as “KEEP AT TERMINATE” is *not* removed as a user attribute of terminated users. This allows the continued processing of activities that had been previously authorized (e.g., for billing purposes, notes, pending orders, or other actions), because the user held the security key.

For example, the PROVIDER security key KEEP AT TERMINATE field is set to **YES** in case a medical order continues to hold an approved status, even though the authorizing provider had been deactivated. As another example, the AudioCare (COTS) pharmacy software depends on the PROVIDER key remaining. The renewal process (OR*3*336, **ORAREN** routine) looks at the original order and creates a new order with the same information, sending an alert to the provider to review and sign the order. If the original provider is no longer active, the order still gets created, but the alert gets forwarded to a surrogate or backup reviewer for signature of the order.

8.12.5.3 SUBORDINATE KEY (Exploding Keys)

If a security key has any associated subordinate keys (i.e., entries in the SUBORDINATE KEY Multiple field), the subordinate keys are automatically assigned along with the overall key. A security key with this feature is called an exploding key, since it and its subordinates are assigned all at once.

NOTE: If entries in the SUBORDINATE KEY Multiple Field are edited, dynamic updating of the security keys already assigned to users does *not* occur.

Exploding security keys *cannot* be exported with software, although, there may be support for this functionality in the future. They are intended to be created by system administrators as a timesaving method in the key allocation process.

8.12.6 Deleting Security Keys

Keys should *not* be deleted from the SECURITY KEY (#19.1) file. Kernel has made the NAME (#.01) field of the SECURITY KEY (#19.1) file uneditable to prevent deletion of security keys through VA FileMan. System administrators should *not* attempt to edit the key global directly to remove a key, since associated pointing relationships are left to cause errors. The one mechanism Kernel does provide for deletion of security keys is through the Kernel Installation and Distribution System (KIDS).

REF: For more information on KIDS, see the "Kernel Installation and Distribution System" section in this manual and the "KIDS Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

8.12.7 Reindexing All Users' Security Keys Option

Figure 120: Reindex the users key's Option

```
SYSTEMS MANAGER MENU ...                                    [EVE]
User Management ...                                       [XUSER]
   Manage User File ...                          [XUSER FILE MGR]
      Reindex the users key's                 [XUSER KEY RE-INDEX]
```

You can use the **Reindex the users key's** [XUSER KEY RE-INDEX] option to re-index all users' security keys in the NEW PERSON (#200) file. If a user has a security key but is lacking the corresponding ^XUSEC cross-reference for the key, you can use this option to regenerate the ^XUSEC cross-reference. While the ^XUSEC cross-reference is being rebuilt, there can be an impact on all users with security key lookups failing in ^XUSEC until the index is entirely rebuilt; therefore, this option should be used with caution and is best delayed until users are *not* signed on.

8.12.8 Using Security Keys with Reverse Locks

If a security key is associated with an option via the REVERSE/NEGATIVE LOCK (#3.01) field, rather than the LOCK (#3) field, it functions to lock out users who hold the key. The security key used for a reverse lock is just like any other key, differing only in the way it is associated with an option. Menu Management’s **Diagram Menus** [XUUSERACC] option indicates the existence of any reverse locks, such as the use of the XMNOPRIV security key to prevent access to MailMan’s shared mail facility.

The typical use of a security key with the REVERSE/NEGATIVE LOCK (#3.01) field is to restrict access to options otherwise available to all users (e.g., MailMan User and other options on the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu [aka **Common** menu]).

8.12.9 Security Key Delegation Levels

Starting with Kernel 8.0, security keys are subject to delegation levels just as options are subject to delegation levels. A field in the NEW PERSON (#200) file, DELEGATION LEVEL, stores a user’s delegation level (for security keys and options). When a security key is delegated, the person to whom it is delegated is assigned a level one number lower than the delegation level of the person doing the delegating. This is to prevent the delegated-to person from removing DELEGATED KEYS from someone with a lower delegation level.

ℹ **REF:** For more information about delegation levels, see the “Secure Menu Delegation” section.

9 Secure Menu Delegation

The job of allocating menu options to users can be a time-consuming activity, so site managers may want to consider delegating this responsibility to application coordinators. Application coordinators are familiar with the menus for their software and can learn how to assign these to new users in their service area.

The **Secure Menu Delegation** [XQSMD MGR] menu allows the site manager to delegate the management of certain menu options to another user (e.g., an application coordinator). This user, now a delegate, can then assign these as primary or secondary options (along with their security keys) to users who fall under their administrative jurisdiction.

For example, the site manager might delegate the management of the Laboratory software options to the Lab Application Coordinator (LAC), and the LAC could then allocate or remove options from everybody in the Laboratory software. The system is set up in such a way that the LAC could also delegate, with the site manager's permission and manager's menu, the management of all the chemistry menus to the head of the Chemistry section, and so on, creating another level of delegation.

There are two divisions in Secure Menu Delegation:

- The menu to create and manage delegates.
- The menu for the delegates themselves to assign options to end users.

9.1 User Interface: Acting as a Delegate

As a delegate, you have been delegated options (usually by system administrators). If you have been delegated options, you can assign these options to computer users on the computer system.

As a delegate, you can assign the following options to your users:

- Options that have been delegated to you.
- Menus that you have created from options delegated to you.
- Options you have created from VA FileMan templates.

As a delegate, you need to understand the basic structure of the OPTION (#19) file, which is a file that points back to itself. That is, a menu is an entry in the OPTION (#19) file; but items on menus are themselves pointers to other entries in the OPTION (#19) file. You should also understand the difference between types of options, be familiar with menu trees, and be sufficiently reluctant to assign great numbers of secondary menus.

9.1.1 Delegate’s Menu

To delegate options to users, you need to be assigned the **Delegate’s Menu Management** [XQSMD USER MENU] menu, which is located under the **Secure Menu Delegation** [XQSMD MGR] menu. The options on the **Delegate’s Menu Management** [XQSMD USER MENU] menu are as shown in Figure 121:

Figure 121: Delegate’s Menu Management Options

```
Delegate’s Menu Management                                  [XQSMD USER MENU]
  Build a New Menu                                         [XQSMD BUILD MENU]
  Edit a User’s Options                                  [XQSMD EDIT OPTIONS]
  Copy Everything About an Option to a New Option                 [XQCOPYOP]
  Copy One Users Menus and Keys to others                   [XQSMD COPY USER]
  Limited File Manager Options (Build)              [XQSMD LIMITED FM OPTIONS]
```

Each of these options on the delegate’s menu is discussed in the topics that follow.

9.1.2 Edit a User’s Options Option

Using the **Edit a User’s Options** [XQSMD EDIT OPTIONS] option allows you to edit a user’s primary and secondary menus. This is the chief method you can use to add (and subtract) options on your users’ menus.

Most of your work is in adding and deleting options on your users’ secondary menus. You are only able to add or delete options from a user’s secondary menu if the option in question has been delegated to you. That means that you do *not* have access to a user’s entire secondary menu; instead, only those options on the secondary menu that are also delegated to you.

If, when you edit a user’s secondary menu, you choose an option that is already on a user’s secondary menu, you are asked if you want to delete it from their secondary menu. Otherwise, you are asked if you want to add the option to their secondary menu.

If you are assigning an option that is locked with a security key, the delegation process checks whether you have been delegated the key as well. If you have, the key is automatically assigned to the user along with the option. If you have *not* been delegated the key, you get an error message saying that you have *not* been delegated the needed security key (the option is assigned to the user, but they do *not* have the key to unlock the option).

If you delete an option that is locked with a security key and that key is delegated to you (and you are at a higher key delegation level than the option holder), the key is deleted along with the option (unless the user holds another option locked by the same security key).

In the example in Figure 122, the user uses the **Edit a User's Options** [XQSMD EDIT OPTIONS] option to add the **LRZ MAIN** menu option to the user's secondary menu. **LRZ MAIN** is locked with a security key and that key is automatically assigned when the option is assigned:

Figure 122: Edit a User's Options—Sample User Dialog

```
Select Delegate's Menu Management Option: EDIT A USER'S OPTIONS

Select NEW PERSON NAME: XUUSER,FIVE
     PRIMARY MENU OPTION: XMUSER// <Enter>      MailMan Menu  .
     No keys needed to delete!.
     No keys needed to give!

     SECONDARY MENU OPTION: LRZ MAIN <Enter>    Lab User Menu  ...
     ZZLRMAIN key also given!

     SECONDARY MENU OPTION: <Enter>

Select NEW PERSON NAME:
```

Unlike secondary menus, you are only able to edit a user's PRIMARY MENU OPTION (#201) field if their current primary menu is an option that has been delegated to you. Otherwise, you are *not* allowed to change that user's PRIMARY MENU OPTION.

NOTE: You *cannot* add or subtract options on a user's primary menu; you can only replace the user's entire PRIMARY MENU OPTION with another one.

9.1.3 Build a New Menu Option

Using the **Build a New Menu** [XQSMD BUILD MENU] option, located on the **Delegate's Menu Management** [XQSMD USER MENU] menu, you can create new menus with menu items chosen from your delegated options.

First, you need to provide an option name for the new menu you are creating. The menu name prefix, used by the delegate to create local options, can be in one of two forms:

- (Preferred) A system administrator-assigned local namespace beginning with the letter "**A**" (e.g., **A6A**).
- (Discouraged) Package namespace (e.g., **LR**) to which the user *must* add the letter "**Z**" (e.g., **LRZ**) in order to avoid conflict with national releases.

NOTE: As of Kernel patch XU*8.0* 482, options in the **A*** namespace can be created *without* adding a "**Z**" to the end of the package namespace.

Once you provide a name for the menu, you are asked to provide the following information:

- Text for the menu.
- Description for the menu.
- Items for the menu (choose from your delegated options).

Once you have created a new menu, you can assign it to your users just as if it were an option delegated to you.

9.1.4 Copy Everything About an Option to a New Option Option

Using the **Copy Everything About an Option to a New Option** [XQCOPYOP] option, you can copy any option on the computer system into a new option. First you are asked which existing option you would like to copy; then, you are asked for a name for the copied option. The option name *must* begin with a namespace assigned to you by the system administrators.

9.1.5 Copy One Users Menus and Keys to others Option

Using the **Copy One Users Menus and Keys to others** [XQSMD COPY USER] option, you can copy the menus and security keys of one user to another user. Each menu or security key you copy, however, *must* have been delegated to you; otherwise, they are skipped in the copy process. What gets copied from one user into the other user are the following fields of the NEW PERSON (#200) file:

- PRIMARY MENU OPTION (#201) (and all descendant menus)
- SECONDARY MENU OPTIONS (#203)
- KEYS (#51)

The PRIMARY MENU OPTION of the user you're copying from *replace***s** the PRIMARY MENU OPTION of the user you are copying to. The SECONDARY MENU OPTIONS (#03) and the KEYS (#51) of the user you are copying from are *merged* into the SECONDARY MENU OPTIONS (#203) and the KEYS (#51) of the user you are copying to.

NOTE: The PSDRPH security key *cannot* be allocated using this option, the **Allocate/De-Allocate of PSDRPH Key (Audited)** [PSO EPCS PSDRPH KEY] option *must* be used to allocate this security key.

9.1.6 Limited File Manager Options (Build) Option

The Secure Menu Delegation system provides a way for delegates to create options out of VA FileMan templates. Delegates who have enough access to VA FileMan to create INPUT, SORT, or PRINT templates can create menu options for their users that directly call these templates.

9.1.6.1 Characteristics of Intended Users

The **Limited File Manager Options (Build)** [XQSMD LIMITED FM OPTIONS] option is designed for delegates, such as some application coordinators who have VA FileMan access to a set of files and can create INPUT, SORT, or PRINT templates. These delegates may have the VA FileMan options for editing or printing without the ability to modify data dictionaries. They may also have explicit file access to a specified set of files via the File Access Management system. Typically they would be working without the special FILE MANAGER ACCESS CODE (#3) field, **DUZ(0)**.

9.1.6.2 System Administrator Setup to Enable Building Options from Templates

To allow a user to create menu options from VA FileMan templates, system administrators *must* first assign to the user:

- **Delegate's Menu Management** [XQSMD USER MENU] menu.
- XQSMDFM Security Key.
- A namespace beginning with the letter **A** (e.g., **A6A**) in which to create options. To do this, use the **Specify Allowable New Menu Prefix** [XQSMD SET PREFIX] option located on the **Secure Menu Delegation** [XQSMD MGR] menu. System administrators are discouraged from assigning package namespaces (e.g., **LR**) to which the user *must* add the letter **Z** (e.g., **LRZ**) to avoid conflict with national releases.

9.1.6.3 Building Options

The tool for building options with VA FileMan templates is called the **Limited File Manager Options (Build)** [XQSMD LIMITED FM OPTIONS]. option It is part of the **Delegate's Menu Management** [XQSMD USER MENU] menu under the **Secure Menu Delegation** [XQSMD MGR] menu and is locked with the XQSMDFM security key.

First, you *must* have created a SORT, PRINT, or INPUT template for a VA FileMan file. Once you have created a template, you can make this template available as an option to your users by turning it into an option.

You can create three types of options:

- Edit-type option (from an EDIT template).
- Print-type option (from PRINT and SORT templates).
- Inquire-type option (from either a PRINT template or a file name).

Once you have turned the template into an option, you can assign that option to your users as you deem necessary. Then, when a user uses the option, they execute the PRINT, SORT, or INPUT template from which the option was created.

Suppose you have created a PRINT template called LRZ REFERRAL PRINT for the Lab's REFERRAL file. To turn this PRINT template into an Inquire option, use the **Limited File Manager Options (Build)** [XQSMD LIMITED FM OPTIONS] option, as shown in Figure 123:

Figure 123: Limited File Manager Options (Build)—Sample User Dialog

```
Select Delegate's Menu Management Option: LIMITED FILE MANAGER OPTIONS (BUILD)
The menu options you build or edit must begin with the namespace:
          LRZ


The option types that may be built are P(rint), E(dit), and I(nquire), and
you must have a template or templates ready to be included in the option.

Or enter D(elete) to DELETE an option


Select Option Type (P/E/I/D): I
     Enter Print Template Name (Optional): LRZ REFERRAL PRINT

     Option Name: LRZ REFERRAL INQUIRE
  Located in the LR (LAB SERVICE) namespace.
  ARE YOU ADDING 'LRZ REFERRAL INQUIRE' AS A NEW OPTION (THE 996TH)? Y <Enter>
(YES)
   OPTION MENU TEXT: DISPLAY A REFERRAL
MENU TEXT: Display a Referral  Replace <Enter>
DESCRIPTION:
  1> Display Lab Referral entries (option created by LAB ADPAC).
  2> <Enter>
EDIT Option: <Enter>

Select Delegate's Menu Management Option:
```

9.2 System Management: Managing Delegates

The options for creating and managing delegates are on the **Secure Menu Delegation** [XQSMD MGR] menu, which is on the Menu Management menu. Typically, system administrators would be the sole holder of this menu. Table 13 lists the options on this menu:

Table 13: Secure Menu Delegation Menu Options

Option Text [Name]	Function
Select Options to be Delegated [XQSMD ADD]	Delegate options
List Delegated Options and their Users [XQSMD BY OPTION]	Print Report
Print All Delegates and their Options [XQSMD BY USER]	Print Report
Remove Options Previously Delegated [XQSMD REMOVE	Undo Delegation
Replicate or Replace a Delegate [XQSMD REPLICATE]	Copy a Delegate
Show a Delegate's Options [XQSMD SHOW]	Print Report

Option Text [Name]	Function
Delegate's Menu Management [XQSMD USER MENU]	Delegate's menu
Specify Allowable New Menu Prefix [XQSMD SET PREFIX]	Assign namespaces

The main options to create and manage delegates are:

- **Select Options to be Delegated** [XQSMD ADD]
- **Replicate or Replace a Delegate** [XQSMD REPLICATE]

9.2.1 Delegating Options: Select Options to be Delegated Option

To delegate options, use the **Select Options to be Delegated** [XQSMD ADD] option from the **Secure Menu Delegation** [XQSMD MGR] menu. Using this option is a two-step process:

1. Choose the users to whom options are delegated.
2. Choose which options to delegate to that group of users.

You can choose to set up one user or many users as delegates. You can choose one option or a group of options to delegate to them.

You also need to assign (*not* delegate!) the **Delegate's Menu Management** [XQSMD USER MENU] menu to the delegate; this menu gives delegates the means to assign delegated options to users, as shown in Figure 124.

Figure 124: Delegating Options: Select Options to be Delegated Option—Sample User Dialog

```
Select Secure Menu Delegation Option: SELECT OPTIONS TO BE DELEGATED

Enter the name(s) of your delegate(s), one at a time

 Name: XUUSER,THREE

 Name: XUUSER,FOUR

 Name: <Enter>

Enter options you wish to DELEGATE TO these users

Add option(s): XUINQUIRE

Add option(s): XUUSERACC

Add option(s): <Enter>

For the following user(s):

1. XUUSER,THREE
2. XUUSER,FOUR

You will delegate the following options:

XUINQUIRE   Inquire
XUUSERACC   Diagram Menus

 Delegated by XUUSER,FIVE on Jul. 21, 2004  3:55 PM.


Ready to delegate these options to these people? Y// <Enter>

Request to add delegated options has been queued, task # 465,
     named: XUUSER,FIVE adding delegated options.
```

9.2.1.1 Delegating Security Keys

If options that you intend to delegate are locked with security keys, you need to delegate the matching keys to the delegate; otherwise, the delegate is *not* able to assign keys to unlock options they have assigned to their users.

If the option is locked with a security key that you possess, the **Select Options to be Delegated** [XQSMD ADD] option branches you to the Key Management program and lets you allocate the appropriate keys to the delegates you are creating.

However, to assign security keys to users, the delegate *must be delegated* the key. To do that, you need to use the **Key Management** [XUKEYMGMT] menu option, **Delegate keys** [XQKEYDEL] option. This option allows you to delegate security keys to delegates by populating the DELEGATED KEYS Multiple field in their NEW PERSON (#200) file entry. Security keys entered in a delegate’s DELEGATED KEYS Multiple allow them to allocate the entered keys to other users (but *not* themselves).

When a delegate assigns options to a user, they can assign the matching security keys as part of that process. However, as an enhancement to a delegate's ability to work with keys, system administrators can assign the delegate the following options from the **Key Management** [XUKEYMGMT] menu option:

- **Allocation of Security Keys** [XUKEYALL]
- **De-allocation of Security Keys** [XUKEYDEALL]
- **Show the Keys of a Particular User** [XQLISTKEY]

As long as the delegate does *not* hold the XUMGR security key, which allows any key to be allocated, the **Key Management** [XUKEYMGMT] menu option only allow delegates to allocate and de-allocate security keys they have been delegated. Kernel also follows key delegation levels with the **Allocation of Security Keys** [XUKEYALL] and **De-allocation of Security Keys** [XUKEYDEALL] options.

NOTE: Key Management [XUKEYMGMT] menu options *must* be separately assigned; they are *not* a part of the **Delegate's Menu Management** [XQSMD USER MENU] menu.

9.2.1.2 Delegation Level (Options and Keys)

The DELEGATION LEVEL (#19.2) field in the NEW PERSON (#200) file specifies the number of steps that a person is from the original delegation of options by the site manager (whose DELEGATION LEVEL is **0**). Starting with Kernel 8.0, the delegation level is also maintained for the DELEGATED KEYS (#52) field. For instance, if the site manager delegates all laboratory options to the Lab ADP Application Coordinator (ADPAC), then the Lab ADPAC would have a DELEGATION LEVEL of **1**. Should the Lab ADPAC further delegate a set of those options to the Lab Chief of Chemistry, the chief would have a level of **2**, and so on.

The use of levels insures that supervision is *not* compromised such that the lower-level user could alter menus or remove security keys of the higher-level person. No attempt is made to determine who actually works for whom, since that information is *not* available to the software. Delegation chains should therefore be constructed with some care.

To modify the set of options (and accompanying security keys) delegated to a particular person, you *must* have a DELEGATION LEVEL equal to, or less than, the person you are trying to modify. If you create a new delegate by delegating some (or all) of the options delegated to you, that person has a DELEGATION LEVEL equal to your level **+1**.

It may be necessary to modify delegation levels using VA FileMan as the organization's structure changes over time.

9.2.2 Further Delegation

The only way a delegate can delegate, rather than simply assign, options to someone else is if the delegate has access to either of the following options:

- **Select Options to be Delegated** [XQSMD ADD]
- **Replicate or Replace a Delegate** [XQSMD REPLICATE]

These options should only be on the **Secure Menu Delegation** [XQSMD MGR] menu. You should carefully evaluate whether to give this menu to delegates, because it gives them the right to further delegate.

9.2.3 Options too Sensitive to Delegate

Certain options (e.g., Programmer-related options) are considered too sensitive or powerful to be delegated. They are marked as *not* delegable in the OPTION (#19) file, and the Secure MenuMan Delegation software does *not* delegate these options. The traditional methods of assigning these menu options *must* be employed by the Site Manager.

It should be noted that a higher-level option, such as the **Systems Manager Menu** [EVE] menu, would still give the delegate access to lower level options, such as the **Menu Management** [XUMAINT] menu, even though **XUMAINT** is itself marked in the OPTION (#19) file as *non*-delegable. The delegation software does *not* follow the option trees down to insure that options of options are *not* delegable.

CAUTION: It is *highly recommended* that the site manager, information security officer (ISO), or chief system administrator review the options marked as too sensitive to be delegated and, using VA FileMan, add any locally sensitive options to this list.

It is the responsibility of each site to insure that the security of the system is *not* violated.

9.2.4 Replicate or Replace a Delegate Option

You can copy the delegated options of a delegate to another user. Use the **Replicate or Replace a Delegate** [XQSMD REPLICATE] option to do this. The options that you transfer to another user do *not* replace any options the user has been previously delegated. They are added to those options, if any. Like the **Select Options to be Delegated** [XQSMD ADD] option, this option also can branch you to the security key allocation program for the new delegate.

You are also asked if the delegated options should be removed from the original delegate. If you say **NO** (**N**), the original delegate remains a delegate. If you say **YES** (**Y**), all delegated options are removed from the original delegate, who is no longer an active delegate. In order to remove the options from a delegate, however, you *must* have a DELEGATION LEVEL lower than they do.

9.2.5 Remove Options Previously Delegated Option

To simply remove an option from a delegate's list of delegable options, use the **Remove Options Previously Delegated** [XQSMD REMOVE] option:

1. Enter the name or names of the delegates from which you want to remove options.
2. Enter the option or options you want to remove from the specified set of delegates.

You are given a chance to review the choices you made; if you say to proceed, a task is queued that removes the options you selected from the delegates you specified.

9.2.6 Specify Allowable New Menu Prefix Option

Use the **Specify Allowable New Menu Prefix** [XQSMD SET PREFIX] option to assign allowable menu prefixes to your delegates. Your delegates need to be given allowable new menu prefixes if they:

- Build new menus.
- Copy options.
- Create options from VA FileMan templates.

Typically, if your delegate works with one particular software application, you would assign them that software's namespace as an allowable prefix. Options that the delegate creates *must* then be prefixed with that namespace, appended with a Z.

If you do *not* specify an allowable prefix for a delegate, they are *not* able to use the following options:

- **Build a New Menu** [XQSMD BUILD MENU]
- **Copy Everything About an Option to a New Option** [XQCOPYOP]
- **Limited File Manager Options (Build)** [XQSMD LIMITED FM OPTIONS]

You can specify multiple new menu prefixes for a given delegate.

9.2.7 Reports

You can use the following options to generate reports about delegates on your system:

- **List Delegated Options and their Users** [XQSMD BY OPTION]
  (Sort by delegated option.)
- **Print All Delegates and their Options** [XQSMD BY USER]
  (Sort by delegate name.)
- **Show a Delegate's Options** [XQSMD SHOW]
  (Display all delegated options for one delegate.)

10 Alerts

10.1 User Interface

When you receive an alert, something on the computer system is requesting your immediate attention. A software application might issue an alert to one or more users when certain conditions are met (e.g., depleted stock levels or abnormal lab test results).

The first time you reach a menu prompt after receiving a particular alert, the alert's message is displayed to you by the menu system. The alert message is displayed along with a standard notice to select the **View Alerts** [XQALERT] option on the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu) to process the alert (see Figure 125).

When you receive an alert, you should find out what the alert is asking of you and attend to it. This is called processing the alert.

Until you process all unprocessed alerts you receive, you'll be reminded that you have pending alerts each time you are at a menu prompt. You do *not*, however, see the alert message; you only see that the first time you receive an alert and reach the menu prompt.

Figure 125: Alert—Sample User Message

```
Dr. You need to enter a progress note on 'KRNPATIENT,ONE'.
          Enter  ‟VA   VIEW ALERTS     to review alerts


Select Systems Manager Menu Option:
```

10.1.1 Processing Alerts

To process alerts, choose the **View Alerts** [XQALERT] from the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). The **View Alerts** [XQALERT] option presents a list of all pending alerts, numbered consecutively with the most recent alerts listed first, with the exception of ***Critical*** alerts (as of Kernel patch XU*8.0*602):

- Critical alerts move to the top of the list and are shown in reverse video.
- Critical alerts are identified by strings of text contained in the ALERT CRITICAL TEXT (#8992.3) file.

REF: For more information on **Critical** alerts, see Section 10.1.1.1, "Critical Alerts."

Information-only alerts are displayed with the letter "**I**" in front of the alert message. When you process Information-only alerts, all that happens is that they are removed from the pending alerts list. Their only purpose was to send you the one-line alert message.

When you process alerts that are *not* Information-only, processing the alert may send you to a particular option or program. Afterwards, you are returned to the View Alerts screen if more alerts need processing, or back to the menu prompt if no pending alerts remain.

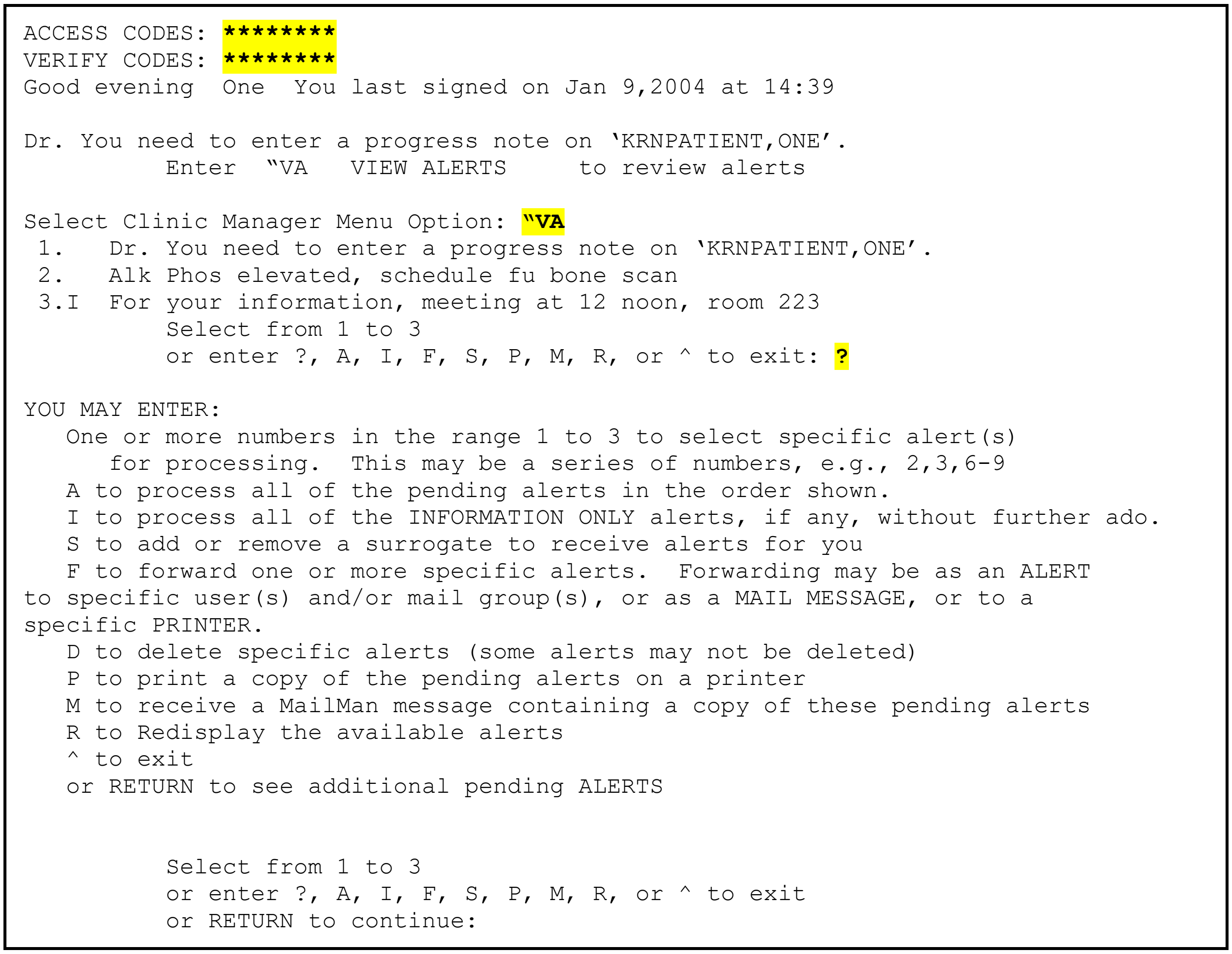
Table 14 lists the various methods for processing alerts from the View Alerts screen. You can enter any of the alert process codes in Table 14 (listed alphabetically):

Table 14: Alert Processing Codes

Process Code	Description
A	Process all alerts in the order shown.
D	Delete specific alerts (some alerts *cannot* be deleted). Only listed if one or more INFORMATION-ONLY alerts have been listed. If unable to delete an alert, users see: “Unable to delete alerts which require action: n,n,n, …”
F	Forward one or more specific alerts. Forwarding may be sent as an alert to specific users or mail groups, a mail message, or sent to a specific printer.
I	Process all INFORMATION-ONLY alerts. Only listed if one or more INFORMATION-ONLY alerts have been listed.
M	List pending alerts in a mail message and deliver the message to your VistA MailMan IN basket.
n	Single number to process a single alert.
n,n,n-n	Range of numbers to process a range of alerts (e.g., 1,3,5-8).
P	Print a copy of the pending alerts to a printer.
R	Redisplay available alerts.
S	Add or remove a surrogate to receive alerts for you. An optional start and end date can also be entered.
^	Exit the alert processing screen by entering a caret (**^**).

The Alert Handler ordinarily deletes alerts once you have processed the alert. If you have processed all pending alerts and try to select the **View Alerts** [XQALERT] option, nothing is displayed. The **View Alerts** option only offers a listing when there are pending alerts; if no alerts are pending, the **View Alerts** option simply returns you to the menu prompt.

Figure 126: View Alerts Option—Sample User Dialog



```
ACCESS CODES: ********
VERIFY CODES: ********
Good evening  One  You last signed on Jan 9,2004 at 14:39

Dr. You need to enter a progress note on ‘KRNPATIENT,ONE’.
          Enter  “VA   VIEW ALERTS     to review alerts

Select Clinic Manager Menu Option: “VA
 1.   Dr. You need to enter a progress note on ‘KRNPATIENT,ONE’.
 2.   Alk Phos elevated, schedule fu bone scan
 3.I  For your information, meeting at 12 noon, room 223
          Select from 1 to 3
          or enter ?, A, I, F, S, P, M, R, or ^ to exit: ?

YOU MAY ENTER:
   One or more numbers in the range 1 to 3 to select specific alert(s)
      for processing.  This may be a series of numbers, e.g., 2,3,6-9
   A to process all of the pending alerts in the order shown.
   I to process all of the INFORMATION ONLY alerts, if any, without further ado.
   S to add or remove a surrogate to receive alerts for you
   F to forward one or more specific alerts.  Forwarding may be as an ALERT
to specific user(s) and/or mail group(s), or as a MAIL MESSAGE, or to a
specific PRINTER.
   D to delete specific alerts (some alerts may not be deleted)
   P to print a copy of the pending alerts on a printer
   M to receive a MailMan message containing a copy of these pending alerts
   R to Redisplay the available alerts
   ^ to exit
   or RETURN to see additional pending ALERTS


          Select from 1 to 3
          or enter ?, A, I, F, S, P, M, R, or ^ to exit
          or RETURN to continue:
```

10.1.1.1 Critical Alerts

The ALERT CRITICAL TEXT file (#8992.3) is a VistA Infrastructure (VI) file designed to accommodate the addition of **Critical** alerts in VistA. This file stores text entries that, when included in an alert, identify it as a **Critical**-type alert.

These file entries are used instead of hard-coded text within the following routines:

- **XQALERT**
- **XQALERT1**

Kernel Patch XU*8.0*690 modified the following reports output:

- **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] option.
- **User Alerts Count Report** [XQAL USER ALERTS COUNT] option.

Any **Critical**-type alerts preceded with the words "**NOT**" or “**NON**”, the only two supported **Critical**-type alert negation indicators, are automatically screened from these reports.

CAUTION: Alerts containing critical text that are *not* to be reported as Critical *must* obey the following rules to pass the Critical-type alert negation test:

- **Only use the negation words “NOT” or “NON”.**
- **Negation words can be upper-, lower-, or mixed-case (i.e., *not* case-sensitive).**
- **Negation words *must* be followed by a single space and no other punctuation marks.**

NOTE: The ALERT CRITICAL TEXT file (#8992.3) file was released with Kernel Patch XU*8.0*513.

As of 04/16/2018, the ALERT CRITICAL TEXT (#8992.3) file contains the following text entries that, when included in an alert, identify it as a **Critical**-type alert:

- **ABNL IMA**

  NOTE: This entry was added with Kernel Patch XU*8.0*690.

- **ABNORMAL IMA**
- **CRITICAL**
- **POSSIBLE MALIG**
- **TESTING-ALERTID-ONLY**

Integration Control Registration (ICR) #6869, ALERT CRITICAL TEXT LOOKUP AND EDIT, is a “Controlled Subscription” type ICR that allows application development teams to release patches that update the ALERT CRITICAL TEXT (#8992.3) file.

Specifically, ICR #6869 grants permission to do the following:

- Look up alert Critical-type text using VA FileMan APIs, such as ^DIC or $$FIND1^DIC.
- Add or edit data in the ALERT CRITICAL TEXT (#8992.3) file using VA FileMan APIs, such as ^DIE, UPDATE^DIE, or FILE^DIE.

CAUTION: Application development teams making changes to the ALERT CRITICAL TEXT (#8992.3) file are responsible for confirming the change does *not* affect Kernel’s reporting of Critical-type alerts.

When adding an entry with Critical-type text to the ALERT CRITICAL TEXT (#8992.3) file, be aware of the following:

- **Reports any alert containing that text as Critical.**
- **Requires careful analysis to confirm changes do *not* cause malfunction of any VistA alerts.**
- **(*recommendation*) Indicate the associated application in the CREATING PACKAGE (#.03) field. Thus, any inquiries regarding the Critical alert text can be directed to the appropriate development team.**
- **Review and determine if the description in the PACKAGE-ID (#.02) field in the ALERT CRITICAL TEXT (#8992.3) file needs to be defined. That field's description indicates that data in this field can further screen alerts from being reported as critical. Its use should be understood when adding entries to the ALERT CRITICAL TEXT (#8992.3) file.**

  If the PACKAGE-ID (#.02) field in the ALERT CRITICAL TEXT (#8992.3) file is defined, then any alert entry in the ALERT (#8992) file containing the Critical text will only be reported as Critical when it's data in the ALERT DATE/TIME sub-file (#8992.01) ALERT ID (#.02) field contains the data string in the PACKAGE-ID (#.02) field in the ALERT CRITICAL TEXT (#8992.3) file.

  In other words, to report an alert as Critical based upon an entry in the ALERT CRITICAL TEXT file when the entry has defined the PACKAGE-ID, the ALERT CRITICAL TEXT file, PACKAGE-ID field data *must* be contained in the ALERT FILE, ALERT DATE/TIME file ALERT ID field.

10.1.2 Deleting Alerts

As of Kernel patch XU*8.0*114, you can delete alerts by using the **D** alert processing code when using the **View Alerts** [XQALERT] option. The user can, if desired, delete specific alerts without viewing or processing them. This option provides the ability to delete "**INFORMATION ONLY**" alerts. Alerts that require processing *cannot* currently be deleted. However, if alerts requiring processing are created with the **XQACNDEL** variable set to **1** they too would be able to be deleted (i.e., the developer of the code that creates the alert can specify if it *must* be processed or can be deleted). Any alerts that were selected for deletion, but could *not* be deleted, are noted for the user.

The ability for the user to delete alerts other than **INFORMATION ONLY** requires that the developers within a software application decide that specific alerts, which would normally invoke processing via an option or routine, can be deleted specifically by the user *without* processing. They would then set the **XQACNDEL** variable to a value of **1** (**one**) prior to calling the SET^XQALERT API to set up the alert. Deletion of an alert by the user (or by system administrators or ADPACs using the existing option) is noted within the ALERT TRACKING (#8992.1) file as deletion by a user (with the user ID) *without* processing of the alert.

10.1.3 Forwarding Alerts

Beginning with Kernel 8.0, you can forward alerts by using the “F” alert processing code when viewing alerts. You can choose one or more alerts and forward them in the following ways:

- Forward as alerts to a specific user on the computer system.
- Forward as alerts to a mail group on the system.
- Copy alerts into mail messages and send to users and mail groups on the system.
- Print to an output device on the system (e.g., a printer).

10.1.4 Surrogates and Alerts

Beginning with Kernel patch XU*8.0*114, you can designate or remove a surrogate for alerts by using the “**S**” alert processing code when viewing alerts. The user can, if desired, specify a start date/time or an end date/time for the surrogate to be effective. If a start date/time is *not* specified, the surrogate becomes active immediately. If an end date/time is specified, the surrogate is removed automatically effective with the first alert sent to the user after the end date/time has passed. If an end date/time is *not* specified, the surrogate is active until another surrogate is specified or the user removes the surrogate.

As of Kernel patch XU*8.0*602, entering a start or end date/time in the past is *not* permitted:

- If a date is entered, then a time is also required.
- If a start date or end date is entered *without* the year and appending the *current* year creates a date in the past, then the next *future* year is appended to the date.

A message is sent to the surrogate to indicate that he has been designated as a surrogate, and a message is sent when the surrogate is removed.

If the user has no alerts and selects the **View Alerts** [XQALERT] option, he is asked if he wants to add or remove a surrogate. The **Alerts - Set/Remove Surrogate for User** [XQALERT SURROGATE SET/REMOVE] option is also provided. It can be used by system administrators or ADPACs to add or remove a surrogate for a selected user. This option is located on the **Alert Management** [XQALERT MGR] menu.

As of Kernel Patch XU*8.0*730, after running the **Alerts - Set/Remove Surrogate for User** [XQALERT SURROGATE SET/REMOVE] option, you can see some examples of surrogate periods for a user, as shown in Figure 127 and Figure 128:

Figure 127: Display of Surrogate Periods—Surrogate with a START DATE and an END DATE

```
Current Surrogate(s):              START DATE               END DATE
1  SURROGATE,ONE                   May 11, 2020@19:31:29    May 18, 2020@00:01
```

Figure 128: Display of Surrogate Periods——Surrogate with a START DATE and No END DATE

```
Current Surrogate(s):              START DATE               END DATE
1  SURROGATE,ONE                   May 11, 2020@19:31:29
```

10.1.4.1 Multiple Sequential Surrogate Periods

Scheduling multiple, sequential, surrogate periods have been available beginning with Kernel Patch XU*8.0*366, but their usage was *not* documented at the time. As of Kernel Patch XU*8.0*730, these surrogate periods are now being documented.

The surrogate periods are sequential; that is, one period ends when the next period starts. Only the last surrogate period has no end date/time specified.

Figure 129 shows an example of creating multiple sequential surrogate periods:

Figure 129: Multiple Sequential Surrogate Periods

```
Select OPTION NAME: XQALERT SURROGATE SET/REMOVE <Enter> Alerts - Set/Remove Surr
ogate for User
Alerts - Set/Remove Surrogate for User
SURROGATE related to which
NEW PERSON entry: USER,ONE
  No current surrogates

Do you want to SET a new surrogate recipient? NO// YES
Select USER to be SURROGATE: SURROGATE,ONE
Enter Date/Time SURROGATE is to start:  (MAY 11, 2020@19:31:24-DEC 31, 2699):
<Enter>
Enter Date/Time SURROGATE is to end:  (MAY 11, 2020@19:31:27-DEC 31, 2699): <Enter>

Current Surrogate(s):                  START DATE                END DATE
1  SURROGATE,ONE                       May 11, 2020@19:31:29

Do you want to SET a new surrogate recipient? NO// YES
Select USER to be SURROGATE: SURROGATE,TWO
Enter Date/Time SURROGATE is to start:  (MAY 11, 2020@19:36:22-DEC 31, 2699): May
18@00:01 <Enter> (MAY 18, 2020@00:01)
Enter Date/Time SURROGATE is to end:  (MAY 11, 2020@19:36:55-DEC 31, 2699): <Enter>

Current Surrogate(s):                  START DATE                END DATE
1  SURROGATE,ONE                       May 11, 2020@19:31:29     May 18, 2020@00:01
2  SURROGATE,TWO                       May 18, 2020@00:01

Do you want to SET a new surrogate recipient? NO// YES
Select USER to be SURROGATE: SURROGATE,THREE
Enter Date/Time SURROGATE is to start:  (MAY 11, 2020@19:44:04-DEC 31, 2699): May
25@00:01 <Enter> (MAY 25, 2020@00:01)
Enter Date/Time SURROGATE is to end:  (MAY 11, 2020@19:44:24-DEC 31, 2699): June
01@00:01 <Enter> (JUN 01, 2020@00:01)

Current Surrogate(s):                  START DATE                END DATE
1  SURROGATE,ONE                       May 11, 2020@19:31:29     May 18, 2020@00:01
2  SURROGATE,TWO                       May 18, 2020@00:01        May 25, 2020@00:01
3  SURROGATE,THREE                     May 25, 2020@00:01        Jun 01, 2020@00:01

Do you want to SET a new surrogate recipient? NO// <Enter>
```

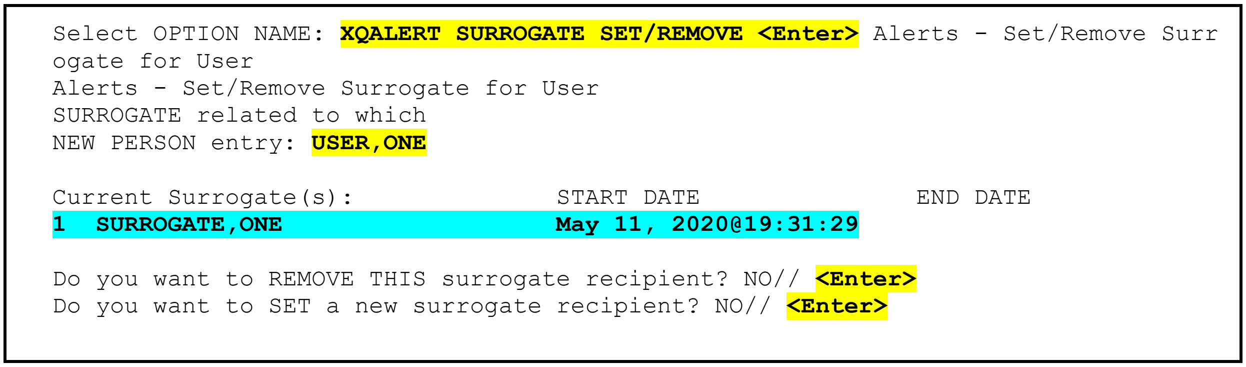
10.1.4.2 Transitive Surrogates

Transitive surrogates occur when a user has a surrogate for a period of time (Surrogate Period 1), and the surrogate also has a surrogate for a period of time (Surrogate Period 2) that coincides with part of the Surrogate Period 1; however, this functionality was broken since Kernel patch XU*8.0*513, where the Surrogate Period 1 was inadvertently removed. This problem has been fixed as of Kernel Patch XU*8.0*730. For examples, see Figure 130 and Figure 131.

10.1.4.2.1 Example 1

Step 1: Parent Surrogate Period: **USER,ONE** has one surrogate period with **SURROGATE,ONE**, as shown in Figure 130:

Figure 130: Transitive Surrogates—Step 1: Parent Surrogate Period

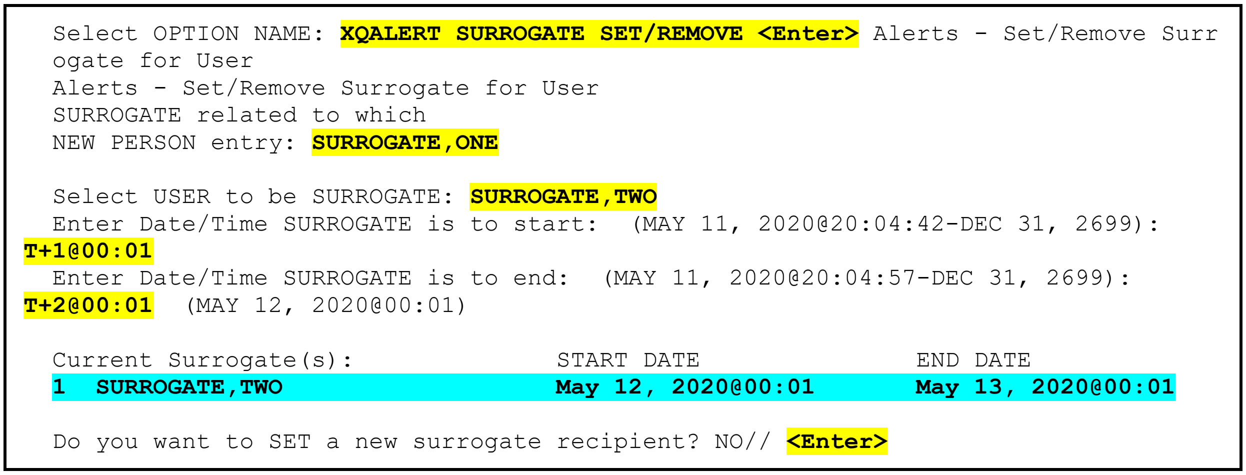


```
  Select OPTION NAME: XQALERT SURROGATE SET/REMOVE <Enter> Alerts - Set/Remove Surr
  ogate for User
  Alerts - Set/Remove Surrogate for User
  SURROGATE related to which
  NEW PERSON entry: USER,ONE

  Current Surrogate(s):                  START DATE                 END DATE
  1  SURROGATE,ONE                       May 11, 2020@19:31:29

  Do you want to REMOVE THIS surrogate recipient? NO// <Enter>
  Do you want to SET a new surrogate recipient? NO// <Enter>
```

10.1.4.2.2 Example 2

Step 2: Child (Transitive) Surrogate Period: **SURROGATE,ONE** has one surrogate period with **SURROGATE,TWO**, as shown in Figure 131:

Figure 131: Transitive Surrogates—Step 2: Child (Transitive) Surrogate Period



```
  Select OPTION NAME: XQALERT SURROGATE SET/REMOVE <Enter> Alerts - Set/Remove Surr
  ogate for User
  Alerts - Set/Remove Surrogate for User
  SURROGATE related to which
  NEW PERSON entry: SURROGATE,ONE

  Select USER to be SURROGATE: SURROGATE,TWO
  Enter Date/Time SURROGATE is to start:  (MAY 11, 2020@20:04:42-DEC 31, 2699):
T+1@00:01
  Enter Date/Time SURROGATE is to end:  (MAY 11, 2020@20:04:57-DEC 31, 2699):
T+2@00:01  (MAY 12, 2020@00:01)

  Current Surrogate(s):                  START DATE                 END DATE
  1  SURROGATE,TWO                       May 12, 2020@00:01         May 13, 2020@00:01

  Do you want to SET a new surrogate recipient? NO// <Enter>
```

10.2 System Management

An alert notifies one or more users of a matter requiring immediate attention. Thus, alerts function as brief notices that are distinct from mail messages or triggered bulletins.

Starting with Kernel 8.0, alerts are stored in the ALERT (#8992) file, which are stored in the **^XTV(8992,** global. Also the ALERT TRACKING (#8992.1) file, stored in **^XTV(8992.1,)** provides a means to track alerts and users' responses to alerts.

For each user to whom an alert is sent, the ALERT TRACKING (#8992.1) file stores the following data:

- Alert name.
- Date created.
- Software identifier of alert.
- User who generated the alert.
- Message text of the alert.
- Action associated with the alert.
- Data associated with the alert.

For each recipient of the alert, the ALERT TRACKING (#8992.1) file stores the following data:

- First date and time observed (shown in menu cycle).
- First date and time selected for processing.
- Date and time processing completed (if any).
- Date and time alert was deleted.
- Forwarding information—If alert was forwarded:
    - User who forwarded it.
    - Date and time of forwarding.

- Surrogate information—If a surrogate was added for alerts:
    - User who was the surrogate.
    - Date and time of the surrogate.

The PATIENT^XQALERT and USER^XQALERT functions provide access to information in the ALERT TRACKING (#8992.1) file.

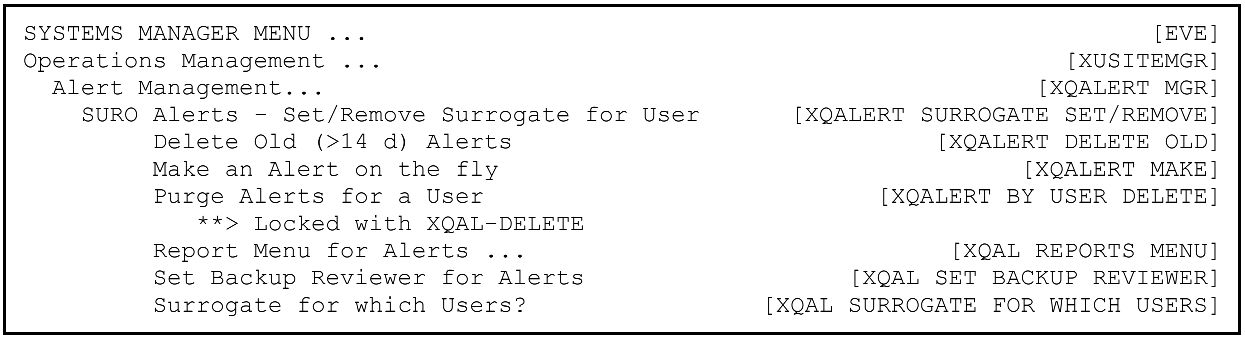
REF: For a description of the XQALERT and other alert-related APIs, see the "Alerts: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet Website.

10.2.1 Alert Management Menu

The **Alert Management** [XQALERT MGR] menu contains the options shown in Figure 132:

Figure 132: Alert Management Menu Options



```
SYSTEMS MANAGER MENU ...                                              [EVE]
Operations Management ...                                       [XUSITEMGR]
  Alert Management...                                         [XQALERT MGR]
    SURO Alerts - Set/Remove Surrogate for User      [XQALERT SURROGATE SET/REMOVE]
         Delete Old (>14 d) Alerts                          [XQALERT DELETE OLD]
         Make an Alert on the fly                                 [XQALERT MAKE]
         Purge Alerts for a User                        [XQALERT BY USER DELETE]
            **> Locked with XQAL-DELETE
         Report Menu for Alerts ...                          [XQAL REPORTS MENU]
         Set Backup Reviewer for Alerts               [XQAL SET BACKUP REVIEWER]
         Surrogate for which Users?                 [XQAL SURROGATE FOR WHICH USERS]
```

10.2.1.1 Alerts - Set/Remove Surrogate for Users Option

The **Alerts - Set/Remove Surrogate for User** [XQALERT SURROGATE SET/REMOVE] option is provided so that system administrators or ADPAC personnel can do the following:

- Set a surrogate to receive alerts for a user.
- Remove a surrogate from receiving alerts for a user.

The option asks for a user to be selected, then is ready to specify a new surrogate for the selected user, or to remove the current surrogate for that user.

This option is *not* needed by the individual users who may select to name or remove a surrogate as one of the options while processing alerts (or if no alerts are present for the user, as his/her only option on selecting alert processing).

10.2.1.2 Delete Old (>14 d) Alerts Option

The **Delete Old (>14 d) Alerts** [XQALERT DELETE OLD] option performs the following functions:

- Purges unprocessed alerts from the ALERT (#8992) file.
- Forwards unprocessed alerts to supervisors or surrogates.

NOTE: Since Kernel Patch XU*8.0*772, entries in the ALERT TRACKING (#8992.1) file are *not* deleted but retained for litigation hold.

You can use the **Delete Old (>14 d) Alerts** [XQALERT DELETE OLD] option to purge all alerts that have been unprocessed for longer than a specified retention period (the default is **14** days.) It is assumed that an alert becomes obsolete within this period and can be purged by system administrators. This option also performs additional functions, which are described below.

This option can be run either directly or as a queued job. You can specify a retention period other than the **14**-day default when you queue the option only, by using the TASK PARAMETERS field of the OPTION SCHEDULING (#19.2) file. If you put a numeric value in the TASK PARAMETERS field of the OPTION SCHEDULING (#19.2) file, this value replaces the default alert retention value of **14** days.

The **Delete Old (>14 d) Alerts** [XQALERT DELETE OLD] option also purges the ALERT TRACKING (#8992.1) file. It purges all entries in the ALERT TRACKING (#8992.1) file that are more than 30 days old. The only exception is if, when an alert is created, the call to create the alert specified a retention period different than 30 days; in this case, the different period is used.

Finally, this option forwards unprocessed alerts to supervisors and surrogates (if this was requested when the alert was created). However, if the period to wait before forwarding exceeds the purging retention period used by this option, the alerts are purged rather than forwarded.

Due to the number of tasks performed by this option, it should be queued through TaskMan on a regular basis. The suggested scheduling frequency is once every day.

10.2.1.3 Make an Alert on the Fly Option

The **Make an Alert on the Fly** [XQALERT MAKE] option allows you to generate an alert on the fly. It interactively asks you for the alert message, recipients, and alert action, if any (you can specify an alert action type of routine or option). It then generates the alert on the fly.

This option is *recommended* primarily for system administrators and ADPACs; it may or may not be appropriate for other selected users.

NOTE: This option does *not* allow the user to set the CAN DELETE WITHOUT PROCESSING (#.1) field in the ALERT (#8992) file.

10.2.1.4 Purge Alerts for a User Option

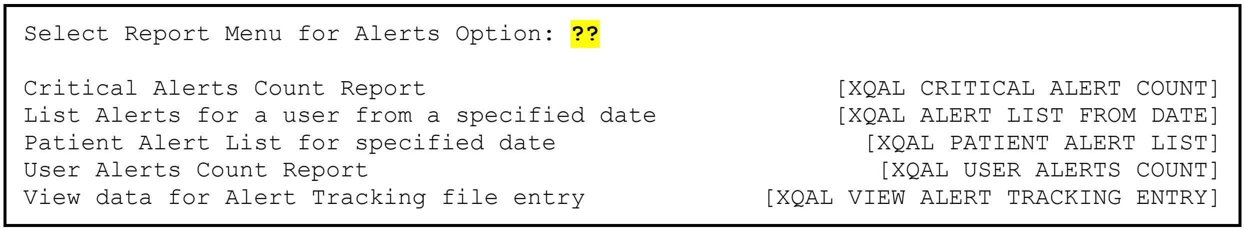
The **Purge Alerts for a User** [XQALERT BY USER DELETE] option allows you to delete alerts for a user. The main purpose of this option is to provide a way to delete alerts for a user who has been inactive for a period of time (e.g., on leave), and who has accumulated a number of alerts that should *not* need processing.

This option is locked with the XQAL-DELETE security key, and it should only be used by system administrators or ADPACs.

10.2.1.5 Report Menu for Alerts Menu

The **Report Menu for Alerts** [XQAL REPORTS MENU] menu provides several options for generating reports on alerts for users or patients. It consists of the submenu items shown in Figure 133:

Figure 133: Report Menu for Alerts Menu Options



```
Select Report Menu for Alerts Option: ??

Critical Alerts Count Report                          [XQAL CRITICAL ALERT COUNT]
List Alerts for a user from a specified date          [XQAL ALERT LIST FROM DATE]
Patient Alert List for specified date                   [XQAL PATIENT ALERT LIST]
User Alerts Count Report                                 [XQAL USER ALERTS COUNT]
View data for Alert Tracking file entry          [XQAL VIEW ALERT TRACKING ENTRY]
```

10.2.1.5.1 Critical Alerts Count Report Option

The **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] option is used to generate a report of users who have alerts defined as **Critical** based upon inclusion of text entries from the ALERT CRITICAL TEXT (#8992.3) file between the specified start and end dates. For example, **Critical**-type alerts contain the following words:

- **ABNL IMA**

  NOTE: This entry was added with Kernel Patch XU*8.0*690.

- **ABNORMAL IMA**
- **CRITICAL**
- **POSSIBLE MALIG**

How the report is presented depends on the order by which method the user selects:

- **Name**—Report lists items alphabetized by name.
- **Number**—Report list items in descending order for the number of **Critical**-type alerts present.

Kernel Patch XU*8.0*690 modified the **Critical Alerts Count Report** output, so any **Critical**-type alerts preceded with the words "**NOT**" or “**NON**”, the only two supported **Critical**-type alert negation indicators, are automatically screened from this report.

CAUTION: Alerts containing critical text that are *not* to be reported as Critical *must* obey the following rules to pass the Critical-type alert negation test:

- **Only use the negation words “NOT” or “NON”.**
- **Negation words can be upper-, lower-, or mixed-case (i.e., *not* case-sensitive).**
- **Negation words *must* be followed by a single space and no other punctuation marks.**

REF: For more information on **Critical**-type alerts, see Section 10.1.1.1, “Critical Alerts.”

For each user who has the specified number of **Critical**-type alerts or more, the report includes the following:

- **Name**—User name.
- **Service/Section**—Section/Service for the user.
- **Alerts**—Number of alerts in the ALERT (#8992) file.
- **Last Sign-on**—Last sign-on date.
- **CRIT**—Number of alerts with **Critical**-type text.
- **Alert**—Date of the oldest alert.

10.2.1.5.1.1 Error Handling—Missing SERVICE/SECTION Data

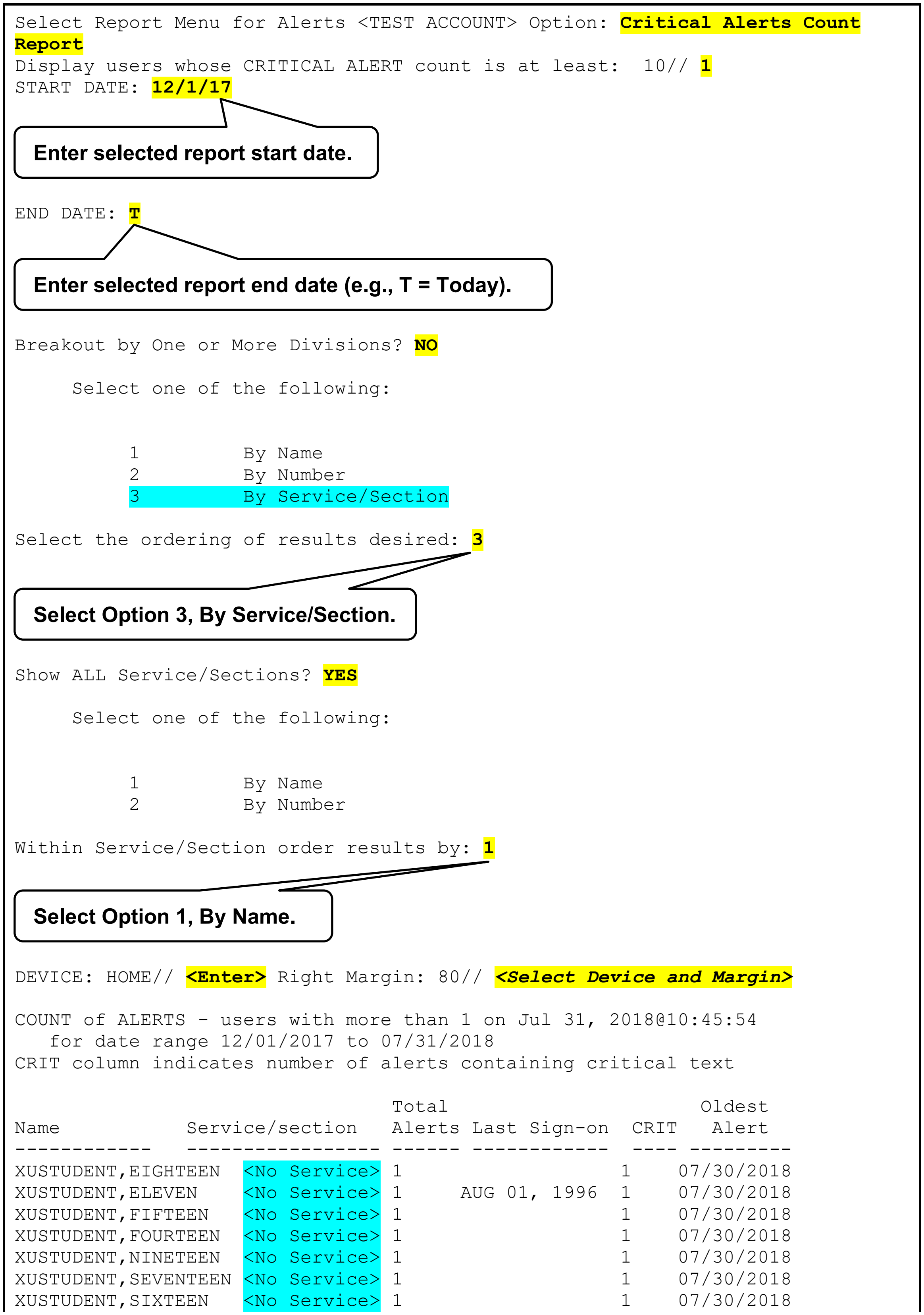
As of Kernel Patch XU*8.0*690, the **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] and **User Alerts Count Report** [XQAL USER ALERTS COUNT] Kernel alert report options no longer abort but gracefully handle reporting users that are missing the SERVICE/SECTION (#29) field data in their NEW PERSON (#200) file definitions.

NOTE: The examples in this section apply to both the **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] and **User Alerts Count Report** [XQAL USER ALERTS COUNT] options, since both options use the same Application Programming Interfaces (APIs).

Using either the **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] or **User Alerts Count Report** [XQAL USER ALERTS COUNT] options, the system no longer aborts but lists those entries missing the SERVICE/SECTION (#29) field data. For example:

1. Execute the **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] option, located on the **Report Menu for Alerts** [XQAL REPORTS MENU] menu (Figure 134).
2. Change users with Critical Alerts of at least **10** to **1**.
3. At the “Display users whose CRITICAL ALERT count is at least: 10//” prompt, enter **1**.
4. Enter a start date prior to the date/time of the pending alert for the active user missing SERVICE/SECTION data.
5. Enter an end date following the date/time of the pending alert for the active user missing SERVICE/SECTION data.
6. Do *not* break out by one or more divisions.
7. Order results “**By Service/Section”**.
8. Sub-sort the results “**By Name”**.
9. Notice that the report will list **<No Service>** for users missing SERVICE/SECTION data. If the Kernel System Parameter limiting the number of errors trapped each day is exceeded, a message is included on the report to note the users who will *not* have error trap entries, as shown in Figure 134.

Figure 134: Testing Reports with Missing Service/Section Data—Critical Alerts Count Report [XQAL CRITICAL ALERT COUNT] Option



```
Select Report Menu for Alerts <TEST ACCOUNT> Option: Critical Alerts Count
Report
Display users whose CRITICAL ALERT count is at least:  10// 1
START DATE: 12/1/17
```

Enter selected report start date.

```
END DATE: T
```

Enter selected report end date (e.g., T = Today).

```
Breakout by One or More Divisions? NO

     Select one of the following:


          1         By Name
          2         By Number
          3         By Service/Section

Select the ordering of results desired: 3
```

Select Option 3, By Service/Section.

```
Show ALL Service/Sections? YES

     Select one of the following:


          1         By Name
          2         By Number

Within Service/Section order results by: 1
```

Select Option 1, By Name.

```
DEVICE: HOME// <Enter> Right Margin: 80// <Select Device and Margin>

COUNT of ALERTS - users with more than 1 on Jul 31, 2018@10:45:54
   for date range 12/01/2017 to 07/31/2018
CRIT column indicates number of alerts containing critical text

                                 Total                      Oldest
Name           Service/section   Alerts Last Sign-on  CRIT   Alert
------------   ----------------- ------ ------------  ---- ---------
XUSTUDENT,EIGHTEEN  <No Service> 1                   1    07/30/2018
XUSTUDENT,ELEVEN    <No Service> 1     AUG 01, 1996  1    07/30/2018
XUSTUDENT,FIFTEEN   <No Service> 1                   1    07/30/2018
XUSTUDENT,FOURTEEN  <No Service> 1                   1    07/30/2018
XUSTUDENT,NINETEEN  <No Service> 1                   1    07/30/2018
XUSTUDENT,SEVENTEEN <No Service> 1                   1    07/30/2018
XUSTUDENT,SIXTEEN   <No Service> 1                   1    07/30/2018
```

```
XUSTUDENT,THIRTEEN  <No Service> 1                   1    07/30/2018
XUSTUDENT,THIRTY    <No Service> 1                   1    07/30/2018
XUSTUDENT,THIRTYONE <No Service> 1                   1    07/30/2018
      Daily Error Trap limit is 100 errors for users missing SERVICE/SECTION.
Limit Reached.  No more entries will be added for '<No Service>' users today!
XUSTUDENT,TWELVE    <No Service> 1     AUG 02, 1996  1    07/30/2018
XUSTUDENT,TWENTY    <No Service> 1                   1    07/30/2018
XUSTUDENT,28        <No Service> 1                   1    07/30/2018

Type <Enter> to continue or '^' to exit:
```

As you can see in Figure 134, the report runs to completion without aborting; even though there are entries with missing SERVICE/SECTION data for multiple users (i.e., indicated on the report as **<No Service>**). The report option indicates *all* entries that are missing SERVICE/SECTION data. It also writes an entry in the error log for a *fixed number* of entries up to the Daily Error Trap limit.

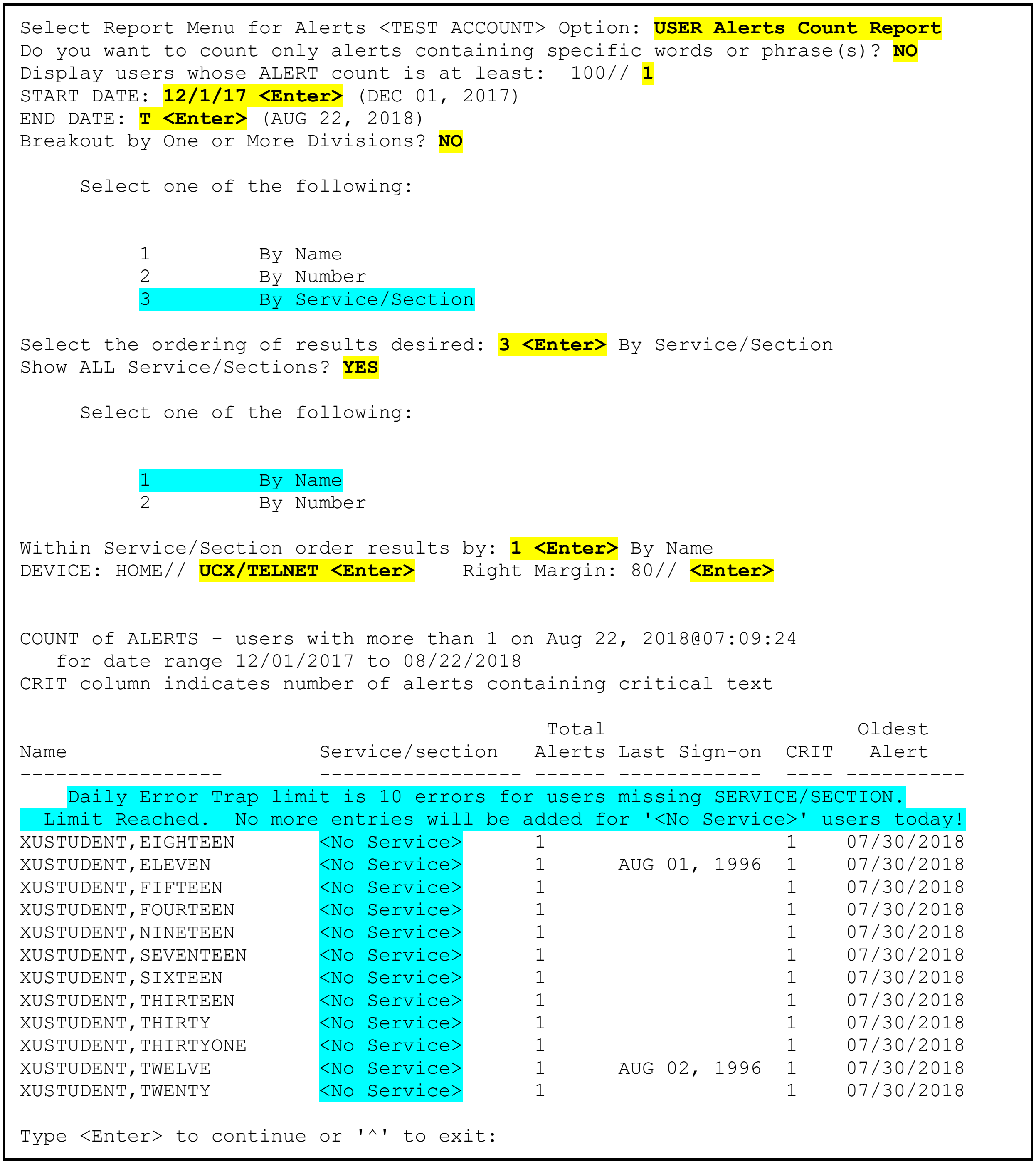
Once that limit has been reached for this error, the option displays a message that no more entries will be added to the error log, as shown in Figure 135:

Figure 135: Sample Error Limit Reached Message for Users Missing SERVICE/SECTION Data

```
Daily Error Trap limit is 100 errors for users missing SERVICE/SECTION.
Limit Reached.  No more entries will be added for '<No Service>' users today!
```

The example in Figure 136 uses the **User Alerts Count Report** [XQAL USER ALERTS COUNT] option:

Figure 136: Testing Reports with Missing Service/Section Data—User Alerts Count Report [XQAL USER ALERTS COUNT] Option



```
Select Report Menu for Alerts <TEST ACCOUNT> Option: USER Alerts Count Report
Do you want to count only alerts containing specific words or phrase(s)? NO
Display users whose ALERT count is at least:  100// 1
START DATE: 12/1/17 <Enter> (DEC 01, 2017)
END DATE: T <Enter> (AUG 22, 2018)
Breakout by One or More Divisions? NO

     Select one of the following:


          1         By Name
          2         By Number
          3         By Service/Section

Select the ordering of results desired: 3 <Enter> By Service/Section
Show ALL Service/Sections? YES

     Select one of the following:


          1         By Name
          2         By Number

Within Service/Section order results by: 1 <Enter> By Name
DEVICE: HOME// UCX/TELNET <Enter>    Right Margin: 80// <Enter>


COUNT of ALERTS - users with more than 1 on Aug 22, 2018@07:09:24
   for date range 12/01/2017 to 08/22/2018
CRIT column indicates number of alerts containing critical text

                                            Total                         Oldest
Name                     Service/section   Alerts Last Sign-on  CRIT   Alert
------------------       ----------------- ------ ------------  ---- ----------
    Daily Error Trap limit is 10 errors for users missing SERVICE/SECTION.
  Limit Reached.  No more entries will be added for '<No Service>' users today!
XUSTUDENT,EIGHTEEN       <No Service>      1                    1     07/30/2018
XUSTUDENT,ELEVEN         <No Service>      1      AUG 01, 1996  1     07/30/2018
XUSTUDENT,FIFTEEN        <No Service>      1                    1     07/30/2018
XUSTUDENT,FOURTEEN       <No Service>      1                    1     07/30/2018
XUSTUDENT,NINETEEN       <No Service>      1                    1     07/30/2018
XUSTUDENT,SEVENTEEN      <No Service>      1                    1     07/30/2018
XUSTUDENT,SIXTEEN        <No Service>      1                    1     07/30/2018
XUSTUDENT,THIRTEEN       <No Service>      1                    1     07/30/2018
XUSTUDENT,THIRTY         <No Service>      1                    1     07/30/2018
XUSTUDENT,THIRTYONE      <No Service>      1                    1     07/30/2018
XUSTUDENT,TWELVE         <No Service>      1      AUG 02, 1996  1     07/30/2018
XUSTUDENT,TWENTY         <No Service>      1                    1     07/30/2018

Type <Enter> to continue or '^' to exit:
```

As you can see in Figure 137, the report runs to completion without aborting; even though there are entries with missing SERVICE/SECTION data for multiple users (i.e., indicated on the report as **<No Service>**). The report option indicates *all* entries that are missing SERVICE/SECTION data. It also writes an entry in the error log for a *fixed number* of entries up to the Daily Error Trap limit.

Once that limit has been reached for this error, the option displays a message that no more entries will be added to the error log. In this example (Figure 137), the limit was reached before the system started listing entries to the report, so the error limit message (Figure 136) appears at the top of the report.

10.2.1.5.2 List Alerts for a user from a specified date Option

The **List Alerts for a user from a specified date** [XQAL ALERT LIST FROM DATE] option reports all alerts from the ALERT TRACKING (#8992.1) file for a selected user within a specified date range. If an end date is *not* specified, the report does *not* run.

The listing includes the following:

- Internal Entry Number (IEN) for the alert in the ALERT TRACKING (#8992.1) file.
- Date and time the alert was generated.
- Message text of the alert.
- Information about any option or routine to be executed for processing the alert.

10.2.1.5.3 Patient Alert List for specified date Option

The **Patient Alert List for specified date** [XQAL PATIENT ALERT LIST] option is used to obtain a list of alerts for a specified patient from the ALERT TRACKING (#8992.1) file for a selected date.

A prompt is provided to obtain a quick scan listing of dates with at least some alerts for the patient on it based on OR and DVB alerts (other patient related alerts need to be identified by looking at each alert’s message text and are included in the full list, but *not* the quick scan).

The listing includes the following:

- Internal Entry Number (IEN) for the alert in the ALERT TRACKING (#8992.1) file.
- Date and time the alert was generated.
- Message text of the alert.
- Information about any option or routine to be executed for processing the alert.

10.2.1.5.4 User Alerts Count Report Option

The **User Alerts Count Report** [XQAL USER ALERTS COUNT] option is used to generate a report on users who have more than a specified number of alerts in the ALERT (#8992) file. This report also includes users who have alerts defined as **Critical** based upon inclusion of text entries from the ALERT CRITICAL TEXT (#8992.3) file. For example, **Critical**-type alerts containing the following words:

- **ABNL IMA**

  NOTE: This entry was added with Kernel Patch XU*8.0*690.

- **ABNORMAL IMA**
- **CRITICAL**
- **POSSIBLE MALIG**

Kernel Patch XU*8.0*690 modified the **User Alerts Count Report** output, so any **Critical**-type alerts preceded with the words "**NOT**" or “**NON**”, the only two supported **Critical**-type alert negation indicators, are automatically screened from this report.

CAUTION: Alerts containing critical text that are *not* to be reported as Critical *must* obey the following rules to pass the Critical-type alert negation test:

- **Only use the negation words “NOT” or “NON”.**
- **Negation words can be upper-, lower-, or mixed-case (i.e., *not* case-sensitive).**
- **Negation words *must* be followed by a single space and no other punctuation marks.**

REF: For more information on **Critical**-type alerts, see Section 10.1.1.1, “Critical Alerts.”

The report covers a specified range of dates. It can be sorted by any of the following data:

- User name.
- Number of alerts.
- Service/Section.

In addition, the report in each of these formats may be generated by Divisions if desired.

For each user who has the specified number of alerts or more, the report includes the following:

- **Name**—User name.
- **Service/Section**—Section/Service for the user.
- **Alerts**—Number of alerts in the ALERTS (#8992) file.
- **Last Sign-on**—Last sign-on date.
- **CRIT**—Number of alerts with **Critical**-type text.
- **Alert**—Date of the oldest alert.

NOTE: For error handling of missing SERVICE/SECTION (#29) field data with the **User Alerts Count Report** [XQAL USER ALERTS COUNT] and **Critical Alerts Count Report** [XQAL CRITICAL ALERT COUNT] options, see Section 10.2.1.5.1.1, "Error Handling—Missing SERVICE/SECTION Data."

10.2.1.5.5 View data for Alert Tracking file entry Option

The **View data for Alert Tracking file entry** [XQAL VIEW ALERT TRACKING ENTRY] option can be used to view data for one or more entries in the ALERT TRACKING (#8992.1) file in captioned format. The internal entry numbers for the entries to be displayed *must* be entered individually.

10.2.1.6 Set Backup Reviewer for Alerts Option

The **Set Backup Reviewer for Alerts** [XQAL SET BACKUP REVIEWER] option provides a mechanism for a user to set entries into the PARAMETERS (#8989.5) file. It assigns an individual as the "Backup Reviewer for Unprocessed Alerts," which is the DISPLAY TEXT (#.02) field for the "XQAL BACKUP REVIEWER" entry in the NAME (#.01) field in the PARAMETER DEFINITION (#8989.51) file, if there is a date specified in the DAYS FOR BACKUP REVIEWER (#.15) field in the ALERT DATE/TIME (#.01) Multiple field in the ALERT (#8992) file for that alert.

If this is the case, an alert that remains unread for the specified number of days is forwarded to the "Backup Reviewer for Unprocessed Alerts" indicated at the lowest level found for processing for the user in the PARAMETERS (#8989.5) file. The following is the processing order (listed lowest to highest level):

1. User
2. OERR Team
3. Team
4. Service
5. Division
6. System

NOTE: This option was released with Kernel patch XU*8.0*174.

10.2.1.7 Surrogate for which Users? Option

The **Surrogate for which Users?** [XQAL SURROGATE FOR WHICH USERS] option provides a view of which users have specified a selected user as surrogates for themselves.

10.2.2 Troubleshooting Using ALERT TRACKING (#8992.1) File

10.2.2.1 Simple Alert to INITIAL RECIPIENT

Figure 137 is an example of a simple alert that was generated by user **SENDER** and delivered to user **RECIPIENT,ORIGINAL**:

- Actual recipient is the one with the **RECIPIENT TYPE** as **INITIAL RECIPIENT**.
- Alert has *not* been viewed nor processed by the recipient.

Figure 137:Sample Output—Simple Alert to INITIAL RECIPIENT

```
NAME: NO-ID;76;3230815.140237           DATE CREATED: AUG 15, 2023@14:02:37
  PKG ID: NO-ID                         GENERATED BY: SENDER
  DISPLAY TEXT: A Simple Alert          ROUTINE TAG: EN
  ROUTINE FOR PROCESSING: AROUTINE
RECIPIENT: RECIPIENT,ORIGINAL
RECIPIENT TYPE: INITIAL RECIPIENT       ALERT DATE/TIME: AUG 15, 2023@14:02:37
```

When the user views and processes the alert, the entry is updated as shown in Figure 138.

NOTE: Typically, it is expected that the time stamp is the same for the following pairs of fields:

- **ALERT FIRST DISPLAYED** and **FIRST SELECTED ALERT**.
- **PROCESSED ALERT** and **DELETED ON**.

Figure 138: Sample Output—When the User Views and Processes the Alert

```
NAME: NO-ID;76;3230815.140237           DATE CREATED: AUG 15, 2023@14:02:37
  PKG ID: NO-ID                         GENERATED BY: SENDER
  DISPLAY TEXT: A Simple Alert          ROUTINE TAG: EN
  ROUTINE FOR PROCESSING: AROUTINE
RECIPIENT: RECIPIENT,ORIGINAL
  ALERT FIRST DISPLAYED: AUG 16, 2023@10:15:02
  FIRST SELECTED ALERT: AUG 16, 2023@10:15:02
  PROCESSED ALERT: AUG 16, 2023@10:17:13
  DELETED ON: AUG 16, 2023@10:17:13
RECIPIENT TYPE: INITIAL RECIPIENT       ALERT DATE/TIME: AUG 15, 2023@14:02:37
```

10.2.2.2 Alert to Surrogate—Who Processes the Alert

When a user has a surrogate and an alert is addressed to the user during the surrogate period, the alert is instead delivered to the surrogate (Figure 139):

- Actual recipient is the one with the **RECIPIENT TYPE** as **INITIAL RECIPIENT-SURROGATE**.
- Alert has *not* been viewed nor processed by any of the recipients.

Figure 139: Sample Output—Alert to Surrogate who Processes the Alert

```
NAME: NO-ID;76;3230728.150546           DATE CREATED: JUL 28, 2023@15:05:46
  PKG ID: NO-ID                         GENERATED BY: SENDER
  DISPLAY TEXT: Alert during surrogacy  ROUTINE TAG: EN
  ROUTINE FOR PROCESSING: AROUTINE
RECIPIENT: RECIPIENT,ORIGINAL
RECIPIENT TYPE: INITIAL RECIPIENT       SENT TO SURROGATE: SURROGATE,ONE
  ALERT DATE/TIME: JUL 28, 2023@15:05:46
RECIPIENT: SURROGATE,ONE
RECIPIENT TYPE: INITIAL RECIPIENT-SURROGATE
  ACTING AS SURROGATE: YES              ALERT DATE/TIME: JUL 28, 2023@15:05:46
SURROGATE FOR: RECIPIENT,ORIGINAL
  DATE/TIME - SURROGATE FOR: JUL 28, 2023@15:05:46
```

It is expected that the surrogate will view and process the alert as shown in Figure 140.

NOTE: The **DATE-TIME RETURNED** field is populated at a later date/time when the surrogate period has ended and either of the two users signs on to CPRS/VistA.

Figure 140: Sample Output—Surrogate Views and Processes the Alert

```
NAME: NO-ID;76;3230728.150546           DATE CREATED: JUL 28, 2023@15:05:46
  PKG ID: NO-ID                         GENERATED BY: SENDER
  DISPLAY TEXT: Alert during surrogacy  ROUTINE TAG: EN
  ROUTINE FOR PROCESSING: AROUTINE
RECIPIENT: RECIPIENT,ORIGINAL
RECIPIENT TYPE: INITIAL RECIPIENT       SENT TO SURROGATE: SURROGATE,ONE
  ALERT DATE/TIME: JUL 28, 2023@15:05:46
RECIPIENT: SURROGATE,ONE
  ALERT FIRST DISPLAYED: JUL 28, 2023@15:09:26
  FIRST SELECTED ALERT: JUL 28, 2023@15:09:26
  PROCESSED ALERT: JUL 28, 2023@15:10:09
  DELETED ON: JUL 28, 2023@15:10:09
RECIPIENT TYPE: INITIAL RECIPIENT-SURROGATE
  ACTING AS SURROGATE: YES              ALERT DATE/TIME: JUL 28, 2023@15:05:46
SURROGATE FOR: RECIPIENT,ORIGINAL
  DATE/TIME - SURROGATE FOR: JUL 28, 2023@15:05:46
  DATE-TIME RETURNED: JUL 28, 2023@15:27:32
```

In this scenario (Figure 140), the Kernel Alerts software is attempting to review all the alerts that occurred during the surrogate period and may attempt to return the alert to the initial recipient if necessary. In this case, the alert is *not* returned, because the surrogate processed the alert.

10.2.2.3 Alert to Surrogate—Who Ignores Alert

If the surrogate does *not* process the alert during the surrogate period, the alert is returned (forwarded) to the initial recipient at the end of the surrogate period and when either of the users signs on to CPRS/VistA, as shown in Figure 141.

Sequence of events:

1. Alert is delivered to surrogate (JUL 28, 2023@09:14:55).
2. Surrogate period ends.
3. One of the recipients signs on to CPRS/VistA.
4. Alert is restored from surrogate (JUL 30, 2023@09:45:21) and forwarded.
5. Original recipient processes the alert (JUL 30, 2023@09:52:12).

Figure 141: Sample Output—Alert to Surrogate Who Ignores Alert

```
NAME: NO-ID;76;3230728.091455           DATE CREATED: JUL 28, 2023@09:14:55
  PKG ID: NO-ID                         GENERATED BY: SENDER
  DISPLAY TEXT: Alert during surrogacy  ROUTINE TAG: EN
  ROUTINE FOR PROCESSING: AROUTINE
RECIPIENT: RECIPIENT,ORIGINAL
  ALERT FIRST DISPLAYED: JUL 30, 2023@09:50:15
  FIRST SELECTED ALERT: JUL 30, 2023@09:50:15
  PROCESSED ALERT: JUL 30, 2023@09:52:12
  DELETED ON: JUL 30, 2023@09:52:12
RECIPIENT TYPE: INITIAL RECIPIENT       SENT TO SURROGATE: SURROGATE,ONE
  ALERT DATE/TIME: JUL 28, 2023@09:14:55
RECIPIENT TYPE: RESTORE FROM SURROGATE  ALERT DATE/TIME: JUL 30, 2023@09:45:21
FORWARDED DATE/TIME: JUL 30, 2023@09:45:21
  FORWARDING CATEGORY: RESTORE FROM SURROGATE
  FORWARDED BY OR FOR: SURROGATE,ONE
  FORWARDING COMMENT: RESTORED FROM SURROGATE
RECIPIENT: SURROGATE,ONE
  ALERT FIRST DISPLAYED: JUL 28, 2023@10:11:03
  FIRST SELECTED ALERT: JUL 28, 2023@10:11:03
  DELETED ON: JUL 30, 2023@09:45:21
RECIPIENT TYPE: INITIAL RECIPIENT-SURROGATE
  ACTING AS SURROGATE: YES              ALERT DATE/TIME: JUL 28, 2023@09:14:55
SURROGATE FOR: RECIPIENT,ORIGINAL
  DATE/TIME - SURROGATE FOR: JUL 28, 2023@09:14:55
  DATE-TIME RETURNED: JUL 30, 2023@09:45:21
```

In this case (Figure 141), note that the **RECIPIENT,ORIGINAL** has **two** instances of the **RECIPIENT TYPE** field:

1. During the surrogate period: **INITIAL RECIPIENT**.
2. After the surrogate period: **RESTORE FROM SURROGATE**.

11 Server Options

11.1 System Management

11.1.1 What is a Server Option?

A server option is a special type of option (stored in the OPTION [#19] file) that can be triggered by mail messages. Addressing a mail message to a server option is termed a “server request.” A server request awakens the option and causes it to execute the following:

- Any M code in the server option’s ENTRY ACTION (#20) field.
- Any M code in the HEADER (#26) field.
- The routine indicated in the ROUTINE (#25) field.
- Any M code in the EXIT ACTION (#15) field.

A server-type option is similar to a run routine-type option. The difference is that a server option is activated by a mail message while a run routine option is activated by a user choosing that option from a menu on a screen. Server options should only be invoked by mail messages (never directly by a user).

The form of the mail message that activates the server option is identical to any other mail message except that it is addressed to **S.<*option name*>**. The “**S.**” (like the “**G.**” form for sending to mail groups) routes the message to the server request software.

11.1.2 What Can Server Options Do?

A server request might trigger a bulletin, send a MailMan reply, or initiate an audit of itself. Developers and local system administrators can also customize the bulletins or MailMan replies.

11.1.3 Can Server Requests Be Denied?

Only server-type options can be activated by mail messages. The following *must* be true for a server request to be processed:

- The server option *must* be set to type “**s**” in the TYPE (#4) field of the OPTION (#19) file. If the type is *not* “**s**” and a request is received, it results in an error that, by default, is recorded in the AUDIT LOG FOR OPTIONS (#19.081) file.
- The server option name *must* be complete and exact when a server request is made, or the request is denied.
- The server option *must not* be disabled (it can be disabled for all requests by setting its LOCK [#3] or OUT OF ORDER MESSAGE [#2] fields).

As long as the conditions listed above are satisfied, the only mechanism a site has for security for server requests is the setting of the server option's SERVER ACTION (#221) field. This field has the settings listed in Table 15:

Table 15: SERVER ACTION (#221) Field Security Values for Server Requests

Value	Description
R	Run immediately. This code causes the server request to be honored in real time as soon as it is received from MailMan (run immediately), provided it is *not* prevented by a setting in the TIMES/DAYS PROHIBITED (#3.91) field.
Q	Queue server. This code causes the server request to be honored (queued) as soon as permitted by the TIMES/DAYS PROHIBITED (#3.91) Multiple field.
N	Notify local authorities. This code causes the server request to create a TaskMan entry but does *not* schedule it to run. A local mail group is notified along with the task number so that it can be approved locally and then scheduled to run using TaskMan's **Requeue Tasks** [XUTM REQ] option.
I	Ignore any server requests. This code causes the software to ignore all requests for this server option. A bulletin or MailMan message can still be sent, however.

When a server request is received, the server option itself is executed similarly to the way a normal option is executed. That is, if a server request causes a server option to be run or queued, the server option, (along with its associated entry action code, header code, routine, and exit action code), does *not* run until the option as a whole runs as scheduled by TaskMan.

11.1.4 How Can the Number of Instances of a Server Option Be Controlled?

To tie a server option to a device of type RESOURCES, use the SERVER DEVICE (#227) field and set the SERVER ACTION (#221) field to **Q** (Queue server) in the OPTION (#19) file. This allows you to control how many instances of the server option can run at any one time. Only as many server option processes can run at any one time as are set up in the associated device's RESOURCE SLOTS (#35) field in the DEVICE (#3.5) file. So if **30** mail messages come in at the same time and attempt to fire off **30** server option processes, you can control the maximum number of simultaneous processes that actually run. Additional server options are able to run when resource slots are freed up from the resource device.

11.1.5 Setting Up a Server Option

A server option has many fields in common with other option types and is set up using the **Edit options** [XUEDITOPT] option, which is locate on the Menu Management [XUMAINT] menu. This option calls the VA FileMan **Template Edit** [DITEMP] option, which prompts for data to be entered in the fields shown in Table 16 (listed in field number order):

Table 16: OPTION (#19) File Field Values When Setting Up a Server Option

Field Name	Description
NAME (#.01)	This should be a namespaced set of **3** to **30** uppercase letters.
MENU TEXT (#1)	Since there is never a menu prompt for a server option, this field should instead contain an accurate description of what this server option does, as it is used by the server request in error messages, bulletins, and MailMan replies. It should be **3** to **50** characters in length.
OUT OF ORDER MESSAGE (#2)	If this field contains between **1** and **80** characters of text, the server option is placed "out of order" and is *not* activated by a server request. The message itself is included in bulletins or MailMan replies that report the failure.
LOCK (#3)	Since server options have no online user associated with them, the existence of a lock in this field prevents the execution of a server option, much like an OUT OF ORDER MESSAGE. The user for all server options is the PostMaster. The originator of a server request is recorded, however, in the return address variable.
DESCRIPTION (#3.5)	This word-processing field should contain an extensive description of the server option intended for the local site manager and system administrators. The description should include an exact description of what the server option does and the resources it requires.
PRIORITY (#3.8)	This field determines the priority at which the server option runs.
TIMES/DAYS PROHIBITED (#3.91) Multiple	This Multiple allows the local system administrators to control the days and times during which the server request is honored. If data is entered that prevents the server option from being honored immediately, the software determines the next available time slice that is *not* prohibited and queues the request for that time. Server options that are marked **R** for Run Immediately in the SERVER ACTION field are instead queued to run at the next non-prohibited time period.
TYPE (#4)	This field *must* always contain the code "**s**" for server-type option, or the request is denied and an error results.

Field Name	Description
EXIT ACTION (#15)	The M code stored in this field is executed just before the server option exits.
ENTRY ACTION (#20)	The M code in this field is executed if the server request is honored. If, as with other options, the variable **XQUIT** exists after the Entry Action is executed, the request is terminated at that point and an error is generated.
ROUTINE (#25)	If there is a routine name in this field in any of the following forms, the routine is run: • ROUTINE • ^ROUTINE • TAG^ROUTINE.
HEADER (#26)	This field of M code is executed, if it exists.
SERVER BULLETIN (#220)	This field is a pointer to the BULLETIN (#3.6) file; it indicates the bulletin to use to notify the local mail group of a server request on their system. If there is no bulletin entered in this field, the default bulletin **XQSERVER** is used. Unless there are pressing reasons to do otherwise, it is recommended that the default bulletin **XQSERVER** be used by leaving the SERVER BULLETIN field blank. If the mail groups pointed to by XQSERVER (or the bulletin pointed to in this field) do *not* contain an active user (i.e., a user possessing a Verify code and no effective TERMINATION DATE) the software turns on auditing (i.e., SERVER AUDIT described below) and sends a MailMan message to the local PostMaster. ⚠ **CAUTION: The most common reason for server options not functioning is that there is no active user associated with the bulletin specified. For security reasons, server options do *not* run without a locally defined active user associated with the chosen bulletin.**
SERVER ACTION (#221)	This SET OF CODES field allows the local system administrators to decide how a server request is to be treated (see Table 15).
SERVER MAIL GROUP (#222)	This field is a pointer to another mail group (the first is pointed to by **XQSERVER** or the bulletin in Field #220) to which server request notifications are to be sent. The software notifies all legitimate users in all mail groups pointed to. It is *recommended* that this field be left blank and a mail group be assigned the chosen bulletin instead.

Field Name	Description
	CAUTION: Server options do *not* work unless there is a local, active user associated with the specified mail group.
SERVER AUDIT (#223)	This field causes the server request to be audited in the AUDIT LOG FOR OPTIONS (#19.081) file. The default is **YES**. The information stored for an audited server option includes: • Option name • User (always PostMaster) • Device • Job number • Date/Time • CPU • Message number • Return address of sender • Subject of the message • Error message A server option can also be audited using the normal option auditing software. Auditing the PostMaster or the namespace "**XQSRV**" captures all server requests.
SUPPRESS BULLETIN (#224)	If set to "**Y**" (**YES**), it prevents a bulletin from being sent under normal conditions. If there is an error or a possible security breach, a bulletin is still fired. If the field is *not* filled in, it takes the default of "**N**," which means that the sending of bulletins is *not* suppressed.
SERVER REPLY (#225)	This SET OF CODES controls the MailMan reply to a server request. The reply is a message returned to the user who has sent the server request and should *not* be confused with the local user to whom the bulletin is addressed. If a reply is requested, the software uses the return address of the sender as supplied by MailMan to send a local or network reply. **REF:** For an example of a server-type option return message, see the Figure 139. The possible codes are: • **N—**No reply is sent (the default).

Field Name	Description
	• **E—**A reply is sent to the return address of the sender only in the event of an error. • **R—**A reply is always sent.
SERVER DEVICE (#227)	Optionally, use this field and the SERVER ACTION (#221) field set to “**Q**” (Queue server) to control the number of server requests for this server option that can be processed at any one time. Enter the name of a device of type RESOURCES (in the DEVICE [#3.5] file). The number of instances of this server option that can run at any one time is limited to the number of resource slots in the selected resource device (i.e., RESOURCE SLOTS (#35) field in the DEVICE (#3.5) file).

11.1.6 Testing if a Site is Reachable: XQSPING Server Option

You can use the **TCP/IP Type Ping Server** [XQSPING] server option to invoke the Kernel **XTSPING** utility at a site. This utility tests to see if the domain to which a message is addressed is reachable. For example, if you want to see if the network link to the Field Office (FO) is working properly, you could address a message to:

```
S.XQSPING@<REDACTED>.VA.GOV
```

If the text of the message and the subject are simply the line “**Testing**”, you should get the message shown in Figure 142 in return:

Figure 142: Sample Message Received when “pinging” a Domain Address

```
MailMan message for Xmuser,One  COMPUTER SPECIALIST
Subj: PING reply to: TESTING [#999] 28 Nov 92 12:17  1 line
From: PING SERVER in ‘IN’ basket.
-----------------------------------------------------------
Testing.
```

The **XTSPING** utility copies the message addressed to it and returns it to the person who sent it.

11.1.7 Testing a Server Option: XQSCHK

You can use the **Server-type Option Test Server** [XQSCHK] server option to return information about a server option on a remote system. You should list the server option you want to test in the text of the message addressed to **XQSCHK**. The subject of the message sent to the **XQSCHK** server option is *not* important. However, the body of the text *must* contain the name of the server option to be checked. When you specify the server option to be checked, do *not* precede the server option name with an "**S.**", instead, list the server option's name exactly as it appears in the OPTION (#19) file's **.01** field.

The **XQSCHK** server option returns Fields #220 to #225 from the OPTION (#19) file to show how the option has been set up. In addition, several other things about the option are investigated and error or warning messages may be also returned.

For example, if you want diagnostic information about a server option named **ZZSERVER**, and the option resides on the system at a field office (FO), you should create a message containing the text **ZZSERVER** and send it to:

```
S.XQSCHK@<REDACTED>.VA.GOV
```

The **XQSCHK** server option unloads the name of the server option (in this example **ZZSERVER**, see Figure 143). Assuming such a server option exists, you would expect to receive a reply in a MailMan message as shown in Figure 143:

Figure 143: XQSCHK Server Option—Sample MailMan Return Message

```
MailMan message for XUUSER,ONE  COMPUTER SPECIALIST
Subj: Server Request Reply from <REDACTED>.VA.GOV
From: Postmaster  in 'IN' basket
--------------------------------------------------------------------------------

                 Nov. 28, 1992  12:18 PM

    Sender: XUUSER,ONE
    Option name: ZZSERVER
    Subject: TESTING XQSCHK
    Message #: 999


    This is a reply from <REDACTED>.VA.GOV
    Checking Server Option ZZSERVER.

    Fields 220 to 225 in the Option File:
           220 - No bulletin selected, will use default XQSERVER.
           221 - The server action code is Run Immediately.
           222 - The mail group ZZGROUP is pointed to.
           223 - Auditing is turned off.
           224 - The server's bulletin is not suppressed.
           225 - Reply mail is sent when an error is trapped.
```

11.1.8 Errors and Warnings from the XQSCHK Server Option

Table 17 lists the errors or warnings that might be included in the return message from the **XQSCHK** server option, along with an explanation of each:

Table 17: XQSCHK Server Option—Error/Warning Messages

Error/Warning Message	Description
`Can't unload name of server from message: [message subject].`	The name of the server option to be tested could *not* be unloaded from the text of the message sent to waken the **XQSCHK** server option. The message should contain just the name of the server option to be tested and nothing more. **XQSCHK** ignores blank lines (up to **4**) and any lines of text that follow the line where it finds the options' name.
`The option [option name] is not in the Option File.`	There is no option in the remote site's OPTION (#19) File that matches the name of the server option that was unloaded from the text of the message. The string it is using to search the OPTION (#19) File is returned in [option name].
`Option [option name] is not shown as a server-type option but a [type].`	The option is *not* marked in the remote OPTION (#19) File as a server-type option, but some other kind of option returned in [type], such as a print-type option.
`[Option name] is marked as Out Of Order with the message: [message].`	The OUT OF ORDER MESSAGE field for that option has been filled in with the text that is returned in [message].
`The expected data in ^DIC(19,[option number], 220) is missing.`	There is no information for this option in Fields #220 through #225. The **220** node of the OPTION (#19) File is missing or blank.
`No bulletin associated with this option default XQSERVER is missing from system.`	There is no bulletin pointed to by Field #220 of this option in the OPTION (#19) File, and the default XQSERVER bulletin has been removed from the system. Server options are *not* run without an associated bulletin, even if it is suppressed.
`Option [option name] points to a bulletin not in the bulletin file.`	WARNING: there is an invalid pointer in Field #220 of the OPTION (#19) File that points to a nonexistent bulletin. The default bulletin XQSERVER is used.

Error/Warning Message	**Description**
`Option [option name] points to a mail group not in the Mail Group File.`	WARNING: there is an invalid pointer in Field #222 of the OPTION (#19) File indicating a mail group that should receive the bulletin in addition to the mail group pointed to by the BULLETIN (#3.6) file.
`There are no mail groups associated with the bulletin [bulletin name].`	The bulletin returned in [bulletin name] does *not* have a mail group associated with it in the BULLETIN (#3.6) file.
`There is no active user associated with the bulletin [bulletin name].`	When following the pointers from the bulletin to the mail group to the NEW PERSON (#200) file, an active user was *not* found. Each server option *must* be linked to a user who has an Access and Verify code and is not terminated.
`There is no routine in field 25 of the Option File for this option.`	This server option has no routine associated with it in the ROUTINE field of the remote site's OPTION (#19) File.
`The routine [routine name] is not on the system.`	The routine that is named in the ROUTINE field of the OPTION (#19) File is *not* found on the system. It has been removed or is in another UCI.
`There is no server action code for this option.`	The required server option action code in Field #221 of the OPTION (#19) File is blank.

12 Help Processor

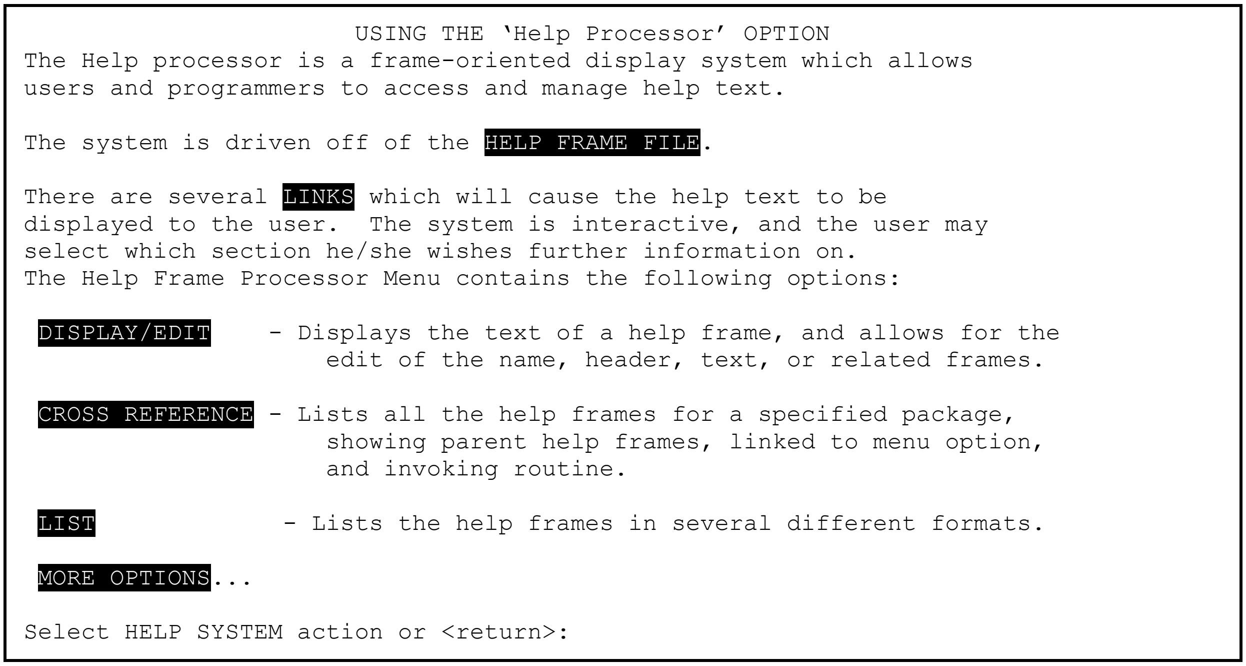
12.1 User Interface

Kernel’s Help Processor is a utility for displaying help frames. A help frame is a screen of text that explains some part of a software application. Each individual help frame can have keyword links to other help frames. Using these keywords, you can navigate through a series of related help frames to learn more about each help frame section.

Some places where you may encounter help frames are:

- When requesting help on options in the menu system.
- When requesting help on a menu in the menu system.
- As a standalone option describing some part of a software application.

Figure 144: Help Frame Example



```
                       USING THE ‘Help Processor’ OPTION
The Help processor is a frame-oriented display system which allows
users and programmers to access and manage help text.

The system is driven off of the HELP FRAME FILE.

There are several LINKS which will cause the help text to be
displayed to the user.  The system is interactive, and the user may
select which section he/she wishes further information on.
The Help Frame Processor Menu contains the following options:

 DISPLAY/EDIT    - Displays the text of a help frame, and allows for the
                     edit of the name, header, text, or related frames.

 CROSS REFERENCE - Lists all the help frames for a specified package,
                     showing parent help frames, linked to menu option,
                     and invoking routine.

 LIST             - Lists the help frames in several different formats.

 MORE OPTIONS...

Select HELP SYSTEM action or <return>:
```

At the bottom of every displayed help frame is a “Select HELP SYSTEM action...” prompt. You have several choices at this prompt. To back your way out of the help frame system, you can simply press the **<Enter>** key. This backs you up one level or exits you if you are at the top level of a help frame tree. If you want to exit quickly from help frames, you can enter **^Q** to quit immediately without having to back all of the way out.

You can list other choices at the “Select HELP SYSTEM action...” prompt by entering a question mark (**?**). The full list of choices are shown in Table 18:

Table 18: Help System Command Actions

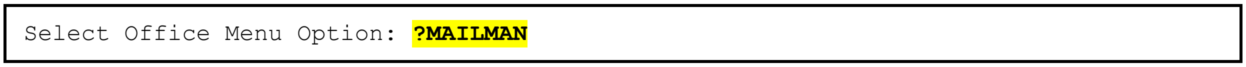
Response	Action
Keyword	Jump to help frame associated with Keyword.
<Enter>	Quit to previous help frame (exit if no previous).
^Q	Quit the help system.
^R	Refresh the current frame.
^T	Table of related frames.
^O	On/off switch for bracketing/reverse video of keywords.
^H	How you got to this frame.
^E	Edit this frame (only if authorized as editor of frame).

Keywords in a help frame are displayed by the help processor in reverse video. If you enter the first few letters of a keyword and press the **<Enter>** key, the help processor jumps to the help frame linked to the entered keyword.

12.1.1 Help Frames in the Menu System

If a menu option has associated help frames, you can display them by entering a question mark (**?**) followed by an option’s menu text or synonym at a menu prompt (i.e., **?**option), as shown in Figure 145:

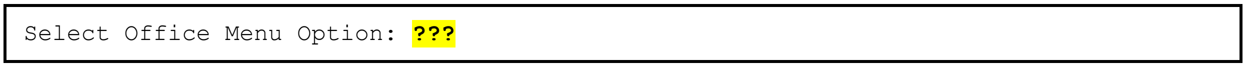
Figure 145: Display a Help Frame for an Option—Entering One Question Mark (?) and Option Name



```
Select Office Menu Option: ?MAILMAN
```

Entering three question marks (**???**) at the menu prompt (Figure 146) indicates which options have associated extended help (help frames).

Figure 146: Display a Help Frame for an Option—Entering Three Question Marks (???)



```
Select Office Menu Option: ???
```

If a menu itself has an associated help frame, entering four question marks (**????**) at the menus "Select ... action: " prompt (Figure 147) displays the help frame associated with that menu if one exists:

Figure 147: Display a Help Frame for an Option—Entering Four Question Marks (????)

```
Select Help Processor Option: ????
```

12.2 System Management

Help frames are entries in the HELP FRAME (#9.2) file. The Header and Text of help frames can be displayed to users to provide instruction about software or other topics. Help frames can be distributed with software or can be created locally to provide information about local policies and procedures.

The options used to create, edit, and link help frames are on the **Help Processor** [XQHELP-MENU] menu, are shown in Figure 148:

Figure 148: Help Processor Menu Options

```
SYSTEMS MANAGER MENU ...                                         [EVE]
Menu Management ...                                          [XUMAINT]
  Help Processor ...                                     [XQHELP-MENU]
     Display/Edit Help Frames                         [XQHELP-DISPLAY]
     List Help Frames                                    [XQHELP-LIST]
     New/Revised Help Frames                           [XQHELP-UPDATE]
     Cross Reference Help Frames                         [XQHELP-XREF]
     Assign Editors                                    [XQHELP-ASSIGN]
     Unassign Editors                                [XQHELP-DEASSIGN]
     Fix Help Frame File Pointers                          [XQHELPFIX]
```

Use of the Help Processor options is explained by help frames associated with the options.

12.2.1 Display/Edit Help Frames Option

The help frames can be displayed with the Display/Edit Help Frames option [XQHELP-DISPLAY]. You can use the **?option** syntax at the select prompt, as shown in Figure 149:

Figure 149: Display/Edit Help Frames Option—Displaying Help Using the ?option Syntax

```
Select Help Processor Option: ?DISPLAY <Enter> /Edit Help Frames
```

12.2.2 List Help Frames Option

The **List Help Frames** [XQHELP-LIST] option can be used to print a series of frames with a table of contents and page numbering to resemble a hard copy manual.

Figure 150: List Help Frames Option—Sample User Dialog

```
Select Help Processor Option: LIST HELP FRAMES
Select primary HELP FRAME from which to list: XUDOC NEW
```

12.2.3 New/Revised Help Frames Option

The **New/Revised Help Frames** [XQHELP-UPDATE] option produces a VA FileMan-generated print of all help frames that have been updated during a specified time period.

12.2.4 Cross Reference Help Frames Option

The **Cross Reference Help Frames** [XQHELP-XREF] option lists any of the following cross-references to a specified set of help frames:

- Parents (other help frames that call the specified help frame).
- Options (options whose HELP FRAME field references the specified help frame).
- Routines (if a developer has entered the routine in the specified help frame's INVOKED BY ROUTINE field).

12.2.5 Fix Help Frame File Pointers Option (Deleting Help Frames)

There is no Kernel utility to delete help frames, but the menu system does *not* generate errors if a pointed-to help frame is missing. If a site chooses to delete help frames using VA FileMan, they should use the **Fix Help Frame File Pointers** [XQHELPFIX] option afterwards to delete dangling pointers from the OPTION file's HELP FRAME field.

12.2.6 Assigning/De-assigning Help Frame Editors

An existing help frame can be edited, through the Help Processor options, by the following people:

- The help frame author.
- Any holder of the XUAUTHOR security key.
- Anyone who has been assigned as an editor to that help frame.

To assign an editor to a given help frame use the **Assign Editors** [XQHELP-ASSIGN] option or to de-assign an editor to a given help frame, use the **Unassign Editor** [XQHELP-DEASSIGN] option.

12.2.7 Disk Space Concerns

Help frames consume disk space. The amount can be considerable if numerous frames are exported with a software application. You can estimate the size of the HELP FRAME (#9.2) file by Kernel's Block Count utility.

Figure 151: Estimating the Size of the HELP FRAME (#9.2) File Using Kernel's Block Count Utility

```
Select Systems Manager Menu Option: PROG <Enter> rammer Options
Select Programmer Options Option: GLOBAL <Enter> Block Count
Block Count for Global ^DIC(9.2)
```

12.2.8 Creating and Editing Help Frames

One way to edit help frames from the HELP FRAME (#9.2) file is to use the **Display/Edit Help Frames** [XQHELP-DISPLAY] option to display the help frame in question. Then, at the "Select Help System Action:" prompt, you can enter **^E** to edit the help frame if you have edit access to the help frame. You have edit access if:

- You are the help frame's author.
- You are assigned as an editor for the help frame.
- You are a holder of the XUAUTHOR security key.

Another handy way to edit help frames is within the help frame system as invoked from a software application. For example, if the help frames are tied to a software's options, you can use the software, invoke the help frame for each field or option, and then edit that help frame on the spot. To edit a help frame in this manner, enter **^E** at the help frame action prompt. To do this, however, you *must* have edit access to the help frame as described above.

12.2.8.1 Namespacing of Help Frames

Like entries in the OPTION (#19) or SECURITY KEY (#19.1) files, entries in the HELP FRAME (#9.2) file *must* be namespaced to avoid overwriting problems.

12.2.8.2 Help Frame Layout Considerations

When entering the text of help frames, you should keep each line to fewer than 80 characters for proper screen display.

NOTE: The text is displayed "as it stands" and is *not* processed by VA FileMan's text formatter. That is, the text is *not* wrapped, and word-processing "windows" are *not* evaluated. Frames are usually **22** lines in length although an end-of-page **READ** is issued to allow a pause if the frame exceeds **22** lines.

If there are only a few lines of text, the Help Processor displays a table at the bottom of the screen of all related frames (those frames that the current frame has keyword links to). The table shows the choices of other frames, so the user need *not* enter the keywords in the text. You can force the table of related frames out of the display by entering enough blank lines so that the frame's length is 20 lines (assuming the display has a page length of **24** lines).

For the Help Processor to identify and highlight keywords, the keywords are entered in the text of the help frame enclosed in square brackets. By convention, keywords in help frames are usually in all capital letters. A square bracket character can be displayed as part of the frame's text by entering two of the characters (e.g., **[[** or **]]**).

If the frames are to be printed using the **List Help Frames** [XQHELP-LIST] option, the resulting help manual has an organized outline, if the frames are linked in a top-down tree structure without any circular connections among the branches.

12.2.8.3 Linking a Help Frame as Help for an Option or Menu

Once a help frame (or a series of help frames) has been created, you can associate it (them) with options by entering the name of the top-level help frame in the HELP FRAME (#3.7) field of the OPTION (#19) file. You can use Menu Manager's **Edit options** [XUEDITOPT] option to do this. That way, when a user enters a single question mark (**?**) in conjunction with the option name, Menu Manager invokes the associated help frame.

Figure 152: Linking Help Frames to an Option—Sample User Dialog

```
Select Systems Manager Menu Option: MENU <Enter> Management
Select Menu Management Option: EDIT OPTIONS
Select OPTION to edit:  XQHELP-MENU <Enter> Help Processor
NAME: XQHELP-MENU// ^HELP FRAME
HELP FRAME: XQHELP
```

13 Error Processing

13.1 User Interface

When an option you are using encounters an error condition, you are usually returned to the menu system. A message is displayed indicating that an error has occurred. You are then presented with the last menu prompt and can continue.

There are certain error conditions, however, that may prohibit or prevent return to the menu system. In these situations, you are halted off the system.

13.2 System Management

The **Error Processing** [XUERRS] menu handles errors for Caché systems. It provides access to options pertaining to the error trap, displaying, printing, and purging errors. Like the error traps provided by the operating systems, the utility allows the investigation of program execution errors or the examination of system errors by capturing a picture of the environment for later reconstruction.

The **%ZTER*** routines are called from ERR^ZU to trap errors and store them in the **^%ZTER** global, a Manager account global that should be translated so that all errors are included on one report. The **XTER*** routines are used to format the error report.

13.2.1 Error Screens

At times you may not want to trap a certain type of error, but merely to count them because you are already aware of the error and can do nothing to prevent it. At other times you may not even want to count the error because it is inevitable or harmless. An error screen is a string of characters that is compared with the error message of every error trapped. Any trapped error whose message contains the screen is screened out. You decide for each screen whether the error is counted or completely ignored. In either case the error is *not* recorded in either the Kernel ERROR LOG (#3.075) file or the TaskMan Error Log. In TaskMan, if a running task encounters a screened error, the submanager still notes the error in the record for that task.

Kernel gives you four options with which to manage your error screens:

- List Error Screens Option [XUTM ERROR SCREEN LIST]
- Add Error Screens Option [XUTM ERROR SCREEN ADD]
- Edit Error Screens Option [XUTM ERROR SCREEN EDIT]
- Remove Error Screens Option [XUTM ERROR SCREEN REMOVE]

NOTE: Even though these four option names are prefixed with "**XUTM**" and located on TaskMan menus, these error screen options apply to all errors and *not* just TaskMan-specific errors. These four options are located on the **Taskman Error Log** [XUTM ERROR] menu, located under the **Taskman Management Utilities** [XUTM UTIL] menu, located under the **Taskman Management** [XUTM MGR] menu, which are all located under the **Systems Manager Menu** [EVE].

13.2.1.1 List Error Screens Option

Figure 153: List Error Screens Option

```
SYSTEMS MANAGER MENU ...                                                       [EVE]
Taskman Management ...                                                    [XUTM MGR]
   Taskman Management Utilities ...                                      [XUTM UTIL]
      Taskman Error Log ...                                             [XUTM ERROR]
         List Error Screens                                 [XUTM ERROR SCREEN LIST]
```

The **List Error Screens** [XUTM ERROR SCREEN LIST] option lists in a simple table the screens you have established and the number of errors that have been screened out by each.

13.2.1.2 Add Error Screens Option

Figure 154: Add Error Screens Option

```
SYSTEMS MANAGER MENU ...                                                       [EVE]
Taskman Management ...                                                    [XUTM MGR]
   Taskman Management Utilities ...                                      [XUTM UTIL]
      Taskman Error Log ...                                             [XUTM ERROR]
         Add Error Screens                                   [XUTM ERROR SCREEN ADD]
```

With the **Add Error Screens** [XUTM ERROR SCREEN ADD] option you can enter a screen and specify whether the errors should be counted. If there are already similar screens in place (e.g., entering SYN when SYNTAX is already established) you are so informed, shown the similar screens, and prompted for confirmation before being asked about the count. Entering two question marks (**??**) at the “Enter Screen To Apply:” prompt displays the list of error screens.

13.2.1.3 Edit Error Screens Option

Figure 155: Edit Error Screens Option

```
SYSTEMS MANAGER MENU ...                                                       [EVE]
Taskman Management ...                                                    [XUTM MGR]
   Taskman Management Utilities ...                                      [XUTM UTIL]
      Taskman Error Log ...                                             [XUTM ERROR]
         Edit Error Screens                                 [XUTM ERROR SCREEN EDIT]
```

Use the **Edit Error Screens** [XUTM ERROR SCREEN EDIT] option if you want to reset the counter on a screen or change your mind about whether or not the screen counts its errors. You *must* type in the exact screen you wish to edit. Again, entering two questions marks displays the list of error screens currently in place.

13.2.1.4 Remove Error Screens Option

Figure 156: Remove Error Screens Option

```
SYSTEMS MANAGER MENU ...                                              [EVE]
Taskman Management ...                                           [XUTM MGR]
   Taskman Management Utilities ...                             [XUTM UTIL]
      Taskman Error Log ...                                    [XUTM ERROR]
         Remove Error Screens                   [XUTM ERROR SCREEN REMOVE]
```

When you type in a screen at the prompt for **Remove Error Screens** [XUTM ERROR SCREEN REMOVE] option, the screen is removed for you. If there are any similar screens, this option asks whether you wish to remove them also. Again, entering two question marks (**??**) displays the list of error screens.

13.2.2 Enhanced Error Processing

Enhanced error processing for Caché sites is supported. Kernel's error trap captures variables in their state at the time errors occur, regardless of how variables may have been **NEW**ed beforehand. Stack levels for the routine call stack are recorded in the error trap in the **$STACK** variable.

The **Error Processing** [XUERRS] menu is comprised of the options that are shown in Figure 157. The description for each option is described in the sections that follow and are arranged in the same order as the options appear on the **Error Processing** menu.

Figure 157: Error Processing Options

```
SYSTEMS MANAGER MENU ...                                              [EVE]
  Programmer Options ...                                           [XUPROG]
    Error Processing ...                                           [XUERRS]
      P1 Print 1 occurrence of each error for T-1 (QUEUE)  [XUERTRP PRINT T-1 1 ERR]
      P2 Print 2 occurrences of errors on T-1 (QUEUED)     [XUERTRP PRINT T-1 2 ERR]
      Clean Error Trap                                    [XUERTRP CLEAN]
      Error Trap Display                                       [XUERTRAP]
      Interactive Print of Error Messages            [XUERTRP PRINT ERRS]
```

13.2.3 Print 1 Occurrence of Each Error for T-1 (QUEUE) Option

The **Print 1 occurrence of each error for T-1 (QUEUE)** [XUERTRP PRINT T-1 1 ERR] option lists the first occurrence of each error recorded on the previous day.

> **T-1** represents "**Today-1 = Yesterday**"

You can queue it to run shortly after midnight:

- If a device is specified, the output is sent to the specified device.
- If a device is *not* specified, the output is placed in a mail message and sent to the individual who queued the option to run. It should be set to automatically requeue at a **1**-day (**D**) interval.

13.2.4 Print 2 Occurrences of Errors on T-1 (QUEUED) Option

The **Print 2 occurrences of errors on T-1 (QUEUED)** [XUERTRP PRINT T-1 2 ERR] option lists the first two occurrences of each error recorded on the previous day:

T-1 represents "**Today-1 = Yesterday**"

It can be queued to run shortly after midnight:

- If a device is specified, the output is sent to the specified device.
- If a device is *not* specified, the output is placed in a mail message and sent to the individual who queued the option to run. It should be set to automatically requeue at a **1**-day (**D**) interval.

13.2.5 Clean Error Trap Option

You can use the **Clean Error Trap** [XUERTRP CLEAN] option to purge the error log. It is locked with the XUPROGMODE security key. You can use the corresponding direct mode utility, ^XTERPUR, in programmer mode. There is also a queueable version, **Error Trap Auto Clean** [XUERTRP AUTO CLEAN] option.

Purging is a partial clearing of the ERROR LOG (#3.075) file stored in the **^%ZTER(1,** global. This global node should *not* be deleted directly since potentially important recent errors would be purged. Deletion of the entire **^%ZTER** global would be a greater mistake since the standard reference data contained in the ERROR MESSAGES (#3.076) file stored in **^%ZTER(2,** would be lost.

You are first prompted for the number of days to leave in the error trap (Figure 158). If you enter a number of days to retain errors, all errors older than the specified number of days are immediately purged:

Figure 158: Choosing the Number of Days to Leave Errors in the Error Trap

```
To Remove ALL entries except the last N days, simply enter the number N at the
prompt.  OTHERWISE, enter return at the first prompt, and a DATE at the second
prompt.  If no ending date is entered at the third prompt, then only the date
specified will be deleted.  If an ending date is entered that range of dates
INCLUSIVE will be deleted from the error log.

Number of days to leave in error trap: 50

          DONE
```

If you just press **<Enter>** instead of entering a number of days to retain, you are then prompted for a start date and end date between which to remove errors (Figure 159). Errors in the period you specify are then purged immediately:

Figure 159: Choosing a Start and End Date Range to Delete Errors from the Error Trap

```
Starting Date to DELETE ERRORS from: 1/1 <Enter> (JAN 01, 2004)
Ending Date to DELETE ERRORS from: 1/31 <Enter> (JAN 31, 2004)
```

The queueable version of this option, **Error Trap Auto Clean** [XUERTRP AUTO CLEAN], can be scheduled to run in the background. By default, it cleans up errors recorded more than **7 days** in the past. You can specify a different interval by placing a numeric value (representing the number of days beyond which to purge) in this option’s TASK PARAMETERS field of the OPTION SCHEDULING (#19.2) file.

13.2.6 Error Trap Display Option

The **Error Trap Display** [XUERTRAP] option displays errors that have been trapped on the system. The messages for these errors are operating-system dependent. You can use the corresponding direct mode utility, ^XTER, from programmer mode.

The error trap tries to capture the following (Figure 160):

- Description of the error.
- Local symbol table.
- Last global reference.
- Other signon statistics.

For Caché, **$ZC** calls are used to record **IO** counts, CPU time, and page faults.

Figure 160: Error Trap Display Option—Sample User Dialog

```
In response to the DATE prompt you can enter:
     ‘S’ to specify text to be matched in error or routine name.

Which date? > T-1
1 error logged on 2/9/95
  1)  <ECODETRAP>PRGMODE+5^%ZOSV:2    07:41:52  KDE,KDE  20801D46
     _TNA4523:

No disconnect error

Which error? >  1

Process ID:  2020107A  (538972282)          JAN 18, 1992 17:19:21

Username: EXAMPLE            Process Name: VISTA User

UCI/VOL: [NXT~NXT~ABC999~NXT:KDAABC999]

$ZA:   0                             $ZB:  \013

Current $IO: _TNA4523:       Current $ZIO: LTA_00129420196A

CPU time:   3.17              Page Faults:       1204

Direct I/O:  81              Buffered I/O:         96

$ZE= <ECODETRAP>PRGMODE+5^%ZOSV:2

 D @XQZ G OUT”

Last Global Ref: ^XUSEC(0,”CUR”,24,2950209.074142)

Which symbol? >
```

Errors can be reported by searching for a date range or character string. Question marks show a count of errors for the selected range:

- Two question marks (**??**) *exclude* disconnects.
- Three question marks (**???**) *include* disconnects.

A string search could be used to find **XQ** in all routines or an **UNDEF** in the definition of all errors. Once an error is identified, the report generator shows the following:

- Job Number
- Username
- **IO** value
- Date/Time
- UCI/Volume Set
- Error Type
- Last Global Reference

- Line of code that caused the error

It then prompts for a listing of variables, enter **^L** to list all or a letter, such as **X**, to list those starting with **X**. The listing can be printed to the screen or to an output device. You can page through the screen listing, one screen at a time, and enter **^Q** to quit or enter ^ to exit at the end of each screen.

A restore feature can be invoked by entering **^R** provided that the user is working in programmer mode. Programmer mode is required as a protection against restoration of variables from within the menu system. To the extent possible, the environment at the time of the error is restored with the routine and local symbol table intact.

Figure 161: Local Symbol Table Help

```
Which symbol? > ?

Enter:
     ^Q to EXIT
     '^' to return to the last question
     Leading character(s) of symbol(s) you wish to examine
     $ to get a display of the $ system variables
     ^L to obtain a list of all symbols
     ^R to restore the symbol table and ... and enter direct mode
```

After reviewing the error log, you are given the opportunity to examine the operating system's error log (Figure 162). Since most VistA applications record their errors in Kernel's error log, there is less need to track VistA errors in the operating system error log.

Figure 162: Choosing to Examine the Operating System's Error Log—Sample User Dialog

```
Do you want to check the OPERATING SYSTEM ERROR TRAP too? NO//
```

13.2.7 Interactive Print of Error Messages Option

The **Interactive Print of Error Messages** [XUERTRP PRINT ERRS] option provides for an interactive print of the first "**n**" of occurrences of an error (where "**n**" is user selectable) over a specified date range.

14 Lock Manager Utility

14.1 Kernel Lock Manager Overview

The Kernel Lock Manager utility is based on the original Class 3 VistA Lock Manager software developed by TM. This software has been updated to Class 1 software via the following Kernel patches:

- **XU*8.0*608—**Contains all the software components that make up the Kernel Lock Manager, which includes the XULM LOCK DICTIONARY (#8993) file.
- **XU*8.0*607—**Populates the XULM LOCK DICTIONARY (#8993) file, which is included in Patch XU*8.0*608. It requires the KIDS enhancement Patch XU*8.0*672.
- **XU*8.0*672—**Enhances the Kernel Installation and Distribution System (KIDS) to allow applications to distribute entries in the XULM LOCK DICTIONARY (#8993) file as KIDS components.

The principle use of the Kernel Lock Manager utility is to assist users in locating locks held by a process that has become dissociated from an active user. Once located, this utility kills the process that owns the lock, thereby releasing the locks held by that process.

The principal advantages of the Kernel Lock Manager utility over the existing Caché utilities include the following functionality:

- Ability to use the Lock Manager from within Veterans Health Information Systems and Technology Architecture (VistA).
- Cross-Node capabilities—No longer need to log into multiple nodes, even if the process that holds the lock is on a different node than the one you currently logged onto. This is accomplished by using the RPC Data Broker to execute Remote Procedure Calls (RPCs) on the other nodes to obtain the lock table and to terminate processes.
- Built-in VistA expertise via the new XULM LOCK DICTIONARY (#8993) file—This file provides in-depth details about the following:
    - Locks
    - Files that the locks reference
    - Processes that hold the locks
- Extendible Lock dictionary—Ability to add information about locks is included in the initial release of the Lock Dictionary. The LOCK TEMPLATE component was added to KIDS via Kernel Patch XU*8.0*672. It allows application developers to add to the Lock Dictionary and distribute their additions via KIDS.

14.2 Configuration

There are two steps to configuring the Lock Manager:

1. Entering Site Parameters
2. Add Lock Manager Users

14.2.1 Entering Site Parameters—Edit Lock Manager Parameters Option

Use the **Edit Lock Manager Parameters** [XULM EDIT PARAMETERS] option to update the Lock Manager parameters in the XULM LOCK MANAGER PARAMETER S (#8993.1) file.

To edit the Lock Manager parameter, perform the following procedure:

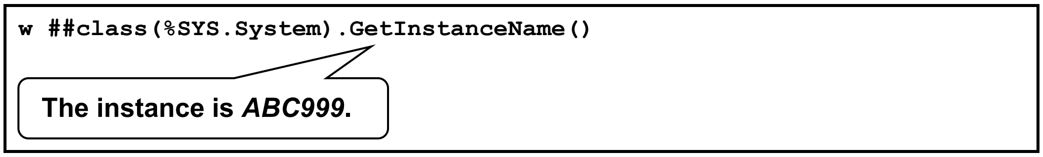
1. From the **Lock Manager Menu** [XULM LOCK MANAGER MENU], select the **Edit Lock Manager Parameters** [XULM EDIT PARAMETERS] option.
2. At the "APPLICATION STATUS:" prompt, set the application status to **ENABLED**.
3. For each node in the system configuration, do the following:
   a. At the "Select NODES:" prompt, enter Caché instance. The name can be obtained by logging onto each node and entering at the M prompt:

      ```
      w ##class(%SYS.System).GetInstanceName()
      ```

      The returned value is the Caché instance name.

      In the example in Figure 163, the instance is named ***ABC999***.

      Figure 163: Sample Code Using GetInstanceName Library Call to Get Instance Name

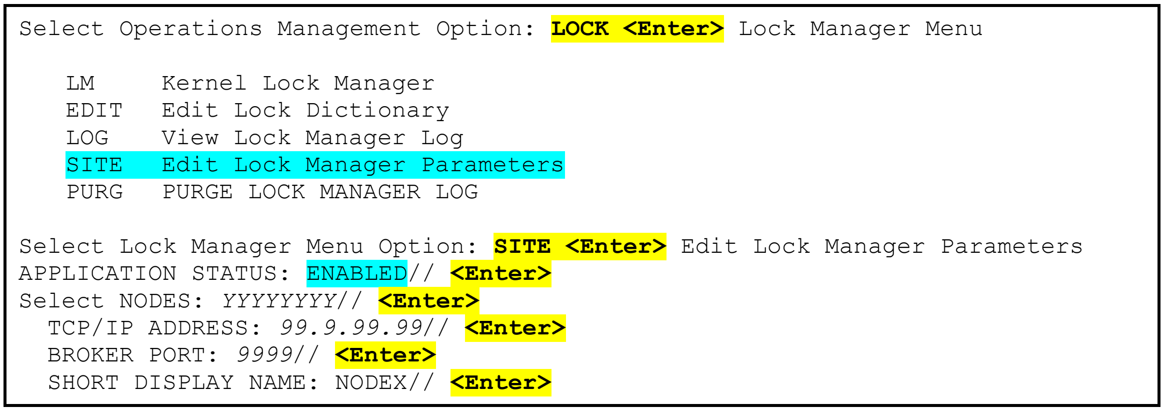
      

      ```
      w ##class(%SYS.System).GetInstanceName()
      ```

      The instance is *ABC999*.

   b. At the "TCP/IP ADDRESS:" prompt, enter the **IP address**.
   c. At the "BROKER PORT:" prompt, enter the **port number of the Broker running on that node**. Either the RPC Broker port or the M-to-M port can be used, but the RPC Broker port is recommended and is more widely available.
   d. The "SHORT DISPLAY NAME" prompt is optional. If the node's name is over **8** characters long, it is necessary at times to display a shortened version. The default is to display only the last **8** characters. If the result is *not* satisfactory, you can enter a shortened name for the node to use as an alternative. **This pertains especially to Linux systems.**

Figure 164: Edit Lock Manager Parameters Option [XULM EDIT PARAMETERS]—Editing Site Parameters



```
Select Operations Management Option: LOCK <Enter> Lock Manager Menu

   LM     Kernel Lock Manager
   EDIT   Edit Lock Dictionary
   LOG    View Lock Manager Log
   SITE   Edit Lock Manager Parameters
   PURG   PURGE LOCK MANAGER LOG

Select Lock Manager Menu Option: SITE <Enter> Edit Lock Manager Parameters
APPLICATION STATUS: ENABLED// <Enter>
Select NODES: YYYYYYYY// <Enter>
  TCP/IP ADDRESS: 99.9.99.99// <Enter>
  BROKER PORT: 9999// <Enter>
  SHORT DISPLAY NAME: NODEX// <Enter>
```

14.2.2 Add Lock Manager Users

The following steps give a user access to the Lock Manager:

1. Assign XULM LOCKS Security Key
2. Edit User Settings for Lock Manager
3. Assign XULM SYSTEM LOCKS Security Key

14.2.2.1 Assign XULM LOCKS Security Key

To assign the XULM LOCKS security key, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **Menu Management** [XUMAINT] menu.
2. At the “Select Menu Management Option:” prompt, select the **Key Management** [XUKEYMGMT] menu.
3. At the “Select Key Management Option:” prompt, select the **Allocation of Security Keys** [XUKEYALL] option.
4. At the “Allocate key:” prompt, enter **XULM LOCKS** security key.
5. At the “Another key:” prompt, press **Enter** to complete your entries.
6. At the “Holder of key:” prompt, enter the user’s name.
7. At the “Another holder:” prompt, enter any additional user names that need access to the Lock Manager. When complete, press **Enter**.
8. At the “You are allocating keys. Do you wish to proceed? YES//” prompt, press **Enter** to accept the **YES** default response.

Figure 165: Adding Lock Manager Users by Assigning XULM LOCKS Security Key

```
Select Systems Manager Menu Option: MENU <Enter> Management

          Edit options
          Key Management ...
          Secure Menu Delegation ...
          Restrict Availability of Options
          Option Access By User
          List Options by Parents and Use
          Fix Option File Pointers
          Help Processor ...
   OPED   Screen-based Option Editor
          Display Menus and Options ...
          Edit a Protocol
          Menu Rebuild Menu ...
          Out-Of-Order Set Management ...
          See if a User Has Access to a Particular Option
          Show Users with a Selected primary Menu

Select Menu Management Option: KEY <Enter> Management

          Allocation of Security Keys
          De-allocation of Security Keys
          Enter/Edit of Security Keys
          All the Keys a User Needs
          Change user’s allocated keys to delegated keys
          Delegate keys
          Keys For a Given Menu Tree
          List users holding a certain key
          Remove delegated keys
          Show the keys of a particular user

Select Key Management Option: ALLOC <Enter> ation of Security Keys

Allocate key: XULM LOCKS

Another key: <Enter>

Holder of key: XUUSER,ONE <Enter>      OX         TECHNICAL WRITER

Another holder: <Enter>

You’ve selected the following keys:

XULM LOCKS

You’ve selected the following holders:

XUUSER,ONE

You are allocating keys.  Do you wish to proceed? YES// <Enter>

XULM LOCKS being assigned to:
     XUUSER,ONE
```

14.2.2.2 Edit User Settings for Lock Manager

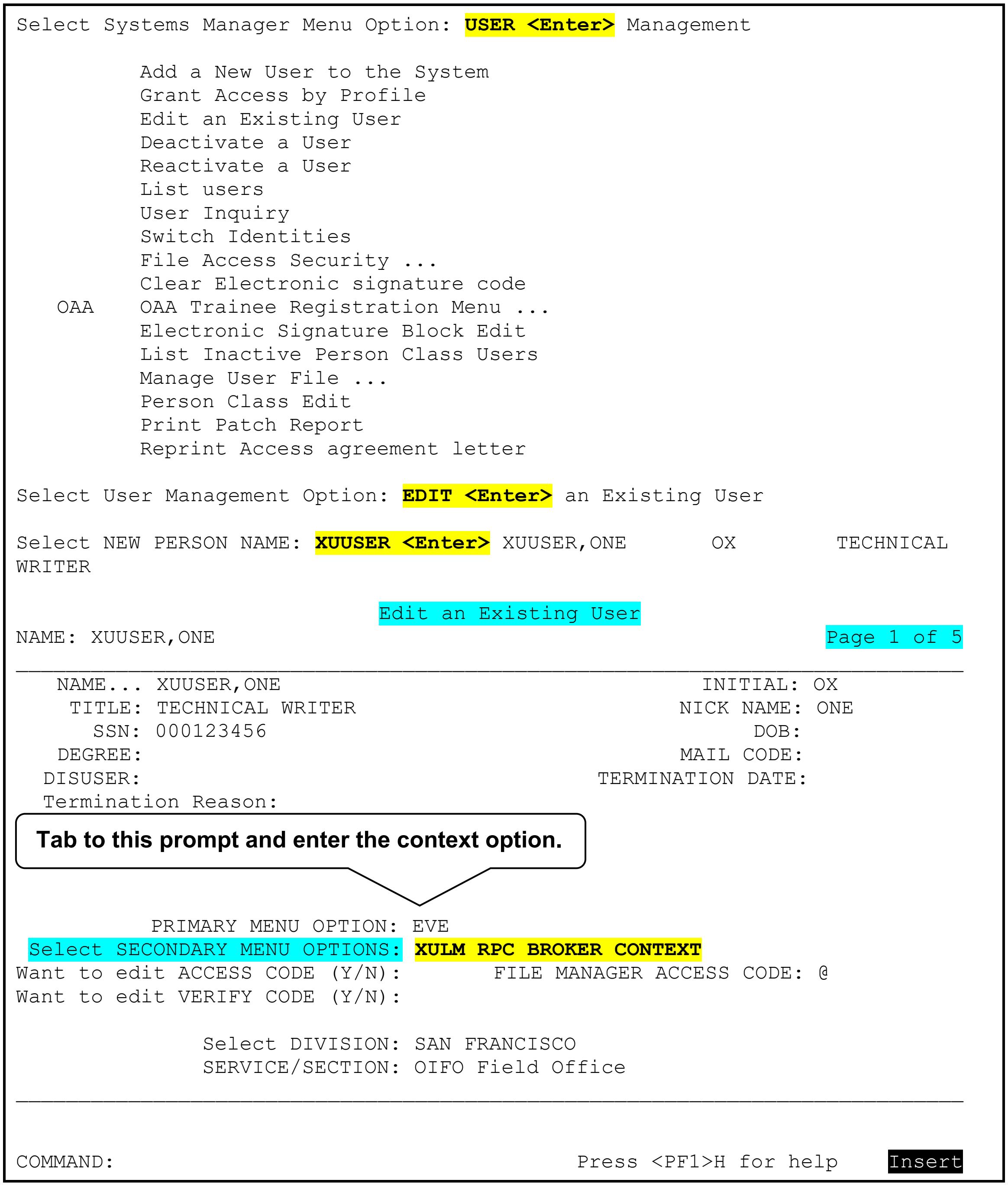
In order to use the Lock Manager each user *must* have the following user settings in the NEW PERSON (#200) file:

- Assign XULM RPC BROKER CONTEXT Option—The **KERNEL LOCK MANAGER** [XULM RPC BROKER CONTEXT] option is the context option the RPC Broker uses for the Lock Manager when making remote procedure calls.
- Set MULTIPLE SIGN-ON Field to **ALLOWED**.

To edit these user settings, do the following:

1. From the **Systems Manager Menu** [EVE], select the **User Management** [XUSER] menu.
2. At the “Select User Management Option:” prompt, select the **Edit an Existing User** [XUSEREDIT] option.
3. At the “Select NEW PERSON NAME:” prompt, enter the user’s name.
4. Assign the **XULM RPC BROKER CONTEXT** option to the user:
    a. In the “Edit an Existing User” main screen (Page 1), tab down to the “Select SECONDARY MENU OPTIONS:” prompt, enter the **XULM RPC BROKER CONTEXT** option (Figure 166).
    b. (Optional) In the “SECONDARY MENU OPTIONS” popup screen, tab to “SYNONYM:” prompt and enter a synonym for this context option (Figure 167).
    c. Tab to the “COMMAND:” prompt, enter **CLOSE**. The “SECONDARY MENU OPTIONS” popup screen closes, and you are returned to the “Edit an Existing User” main screen (Page 1).

Figure 166: Assigning XULM RPC BROKER CONTEXT Option—Sample User Entries and System Responses (1 of 2)

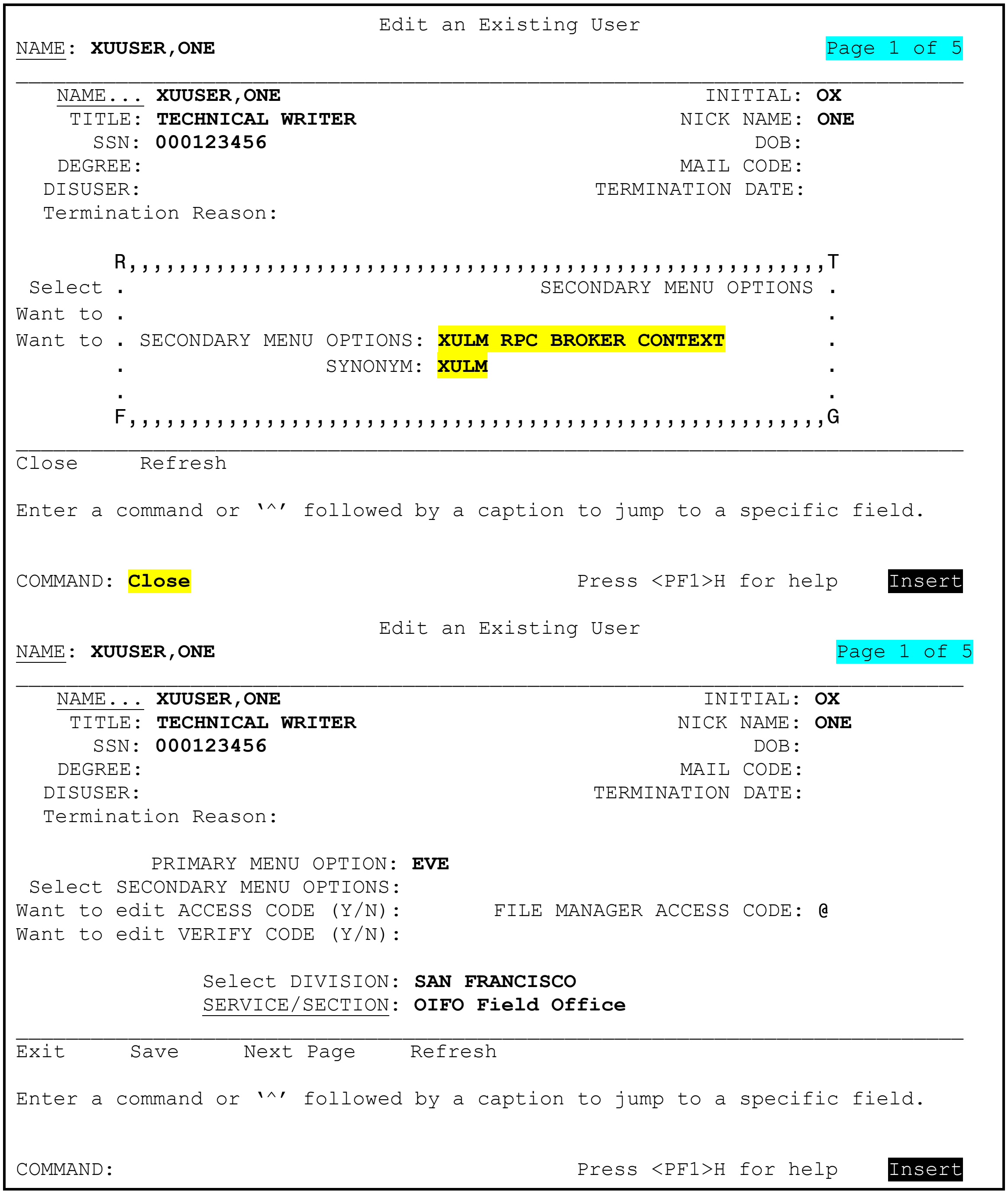


```
Select Systems Manager Menu Option: USER <Enter> Management

          Add a New User to the System
          Grant Access by Profile
          Edit an Existing User
          Deactivate a User
          Reactivate a User
          List users
          User Inquiry
          Switch Identities
          File Access Security ...
          Clear Electronic signature code
   OAA    OAA Trainee Registration Menu ...
          Electronic Signature Block Edit
          List Inactive Person Class Users
          Manage User File ...
          Person Class Edit
          Print Patch Report
          Reprint Access agreement letter

Select User Management Option: EDIT <Enter> an Existing User

Select NEW PERSON NAME: XUUSER <Enter> XUUSER,ONE       OX        TECHNICAL
WRITER

                             Edit an Existing User
NAME: XUUSER,ONE                                                 Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                  INITIAL: OX
    TITLE: TECHNICAL WRITER                          NICK NAME: ONE
      SSN: 000123456                                       DOB:
   DEGREE:                                           MAIL CODE:
  DISUSER:                                    TERMINATION DATE:
  Termination Reason:
```

Tab to this prompt and enter the context option.

```
           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS: XULM RPC BROKER CONTEXT
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office
____________________________________________________________________________


COMMAND:                                     Press <PF1>H for help    Insert
```

Figure 167: Assigning XULM RPC BROKER CONTEXT Option—Sample User Entries and System Responses (2 of 2)

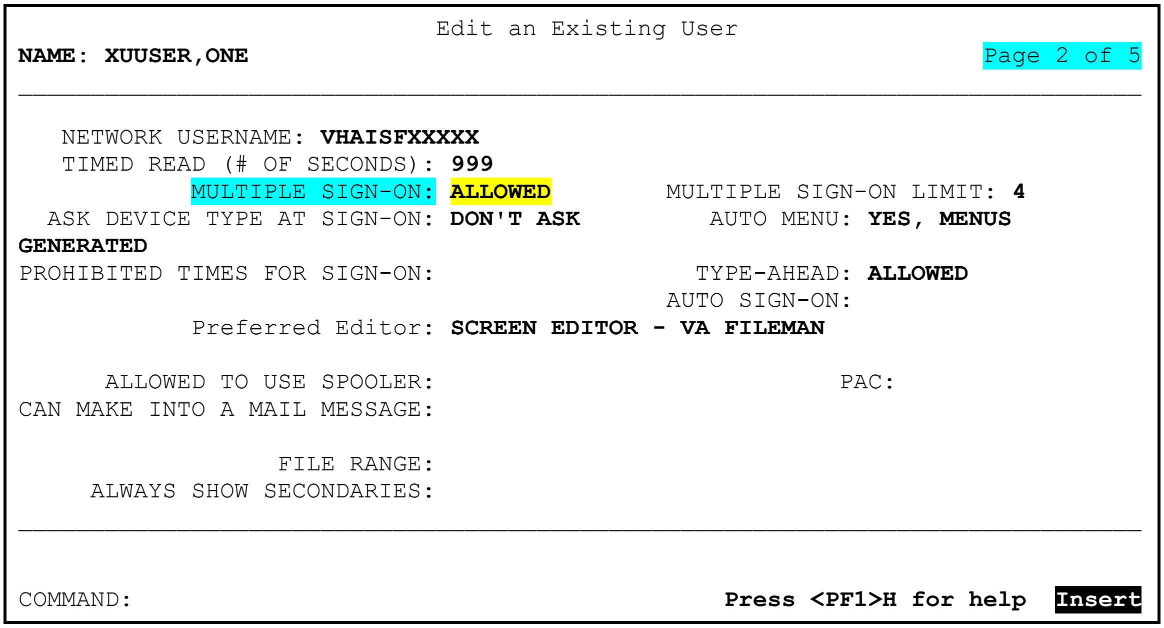


```
                          Edit an Existing User
NAME: XUUSER,ONE                                                 Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                       INITIAL: OX
    TITLE: TECHNICAL WRITER                               NICK NAME: ONE
      SSN: 000123456                                            DOB:
   DEGREE:                                                MAIL CODE:
  DISUSER:                                         TERMINATION DATE:
  Termination Reason:

        R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
 Select .                                 SECONDARY MENU OPTIONS .
Want to .                                                        .
Want to . SECONDARY MENU OPTIONS: XULM RPC BROKER CONTEXT        .
        .                SYNONYM: XULM                           .
        .                                                        .
        F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
____________________________________________________________________________
Close     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND: Close                                  Press <PF1>H for help  Insert
                          Edit an Existing User
NAME: XUUSER,ONE                                                  Page 1 of 5
____________________________________________________________________________
   NAME... XUUSER,ONE                                       INITIAL: OX
    TITLE: TECHNICAL WRITER                               NICK NAME: ONE
      SSN: 000123456                                            DOB:
   DEGREE:                                                MAIL CODE:
  DISUSER:                                         TERMINATION DATE:
  Termination Reason:

           PRIMARY MENU OPTION: EVE
 Select SECONDARY MENU OPTIONS:
Want to edit ACCESS CODE (Y/N):       FILE MANAGER ACCESS CODE: @
Want to edit VERIFY CODE (Y/N):

               Select DIVISION: SAN FRANCISCO
               SERVICE/SECTION: OIFO Field Office
____________________________________________________________________________
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                        Press <PF1>H for help  Insert
```

5. Tab to the "COMMAND:" prompt, enter **NEXT PAGE (N)**. The "Edit an Existing User" screen moves to Page 2.

6. Set the MULTIPLE SIGN-ON Field to **ALLOWED**:
   a. In the "Edit an Existing User" main screen (Page 2), tab down to the "MULTIPLE SIGN-ON" field (Figure 168).
   b. If *not* already set to **ALLOWED**, enter **ALLOWED** at the "MULTIPLE SIGN-ON:" prompt.

Figure 168: Setting MULTIPLE SIGN-ON Field to ALLOWED—Sample User Entries and System Responses



```
                             Edit an Existing User
NAME: XUUSER,ONE                                                   Page 2 of 5
______________________________________________________________________________

   NETWORK USERNAME: VHAISFXXXXX
   TIMED READ (# OF SECONDS): 999
            MULTIPLE SIGN-ON: ALLOWED          MULTIPLE SIGN-ON LIMIT: 4
  ASK DEVICE TYPE AT SIGN-ON: DON'T ASK           AUTO MENU: YES, MENUS
GENERATED
PROHIBITED TIMES FOR SIGN-ON:                    TYPE-AHEAD: ALLOWED
                                               AUTO SIGN-ON:
            Preferred Editor: SCREEN EDITOR - VA FILEMAN

      ALLOWED TO USE SPOOLER:                            PAC:
CAN MAKE INTO A MAIL MESSAGE:

                  FILE RANGE:
     ALWAYS SHOW SECONDARIES:
______________________________________________________________________________


COMMAND:                                         Press <PF1>H for help   Insert
```

14.2.2.3 Assign XULM SYSTEM LOCKS Security Key

CAUTION: Use discretion when assigning this security key; deleting a system lock can result in database corruption!

To assign the XULM SYSTEM LOCKS security key, perform the following procedure:

1. From the **Systems Manager Menu** [EVE], select the **Menu Management** [XUMAINT] menu.
2. At the "Select Menu Management Option:" prompt, select the **Key Management** [XUKEYMGMT] menu.
3. At the "Select Key Management Option:" prompt, select the **Allocation of Security Keys** [XUKEYALL] option.
4. At the "Allocate key:" prompt, enter **XULM SYSTEM LOCKS** security key.
5. At the "Another key:" prompt, press **Enter** to complete your entries.
6. At the "Holder of key:" prompt, enter the user's name.

7. At the “Another holder:” prompt, enter any additional user names that need access to the Lock Manager. When complete, press **Enter**.
8. At the “You are allocating keys. Do you wish to proceed? YES//” prompt, press **Enter** to accept the **YES** default response.

Figure 169: Adding Lock Manager Users by Assigning XULM SYSTEM LOCKS Security Key

```
Select Systems Manager Menu Option: MENU <Enter> Management

          Edit options
          Key Management ...
          Secure Menu Delegation ...
          Restrict Availability of Options
          Option Access By User
          List Options by Parents and Use
          Fix Option File Pointers
          Help Processor ...
   OPED   Screen-based Option Editor
          Display Menus and Options ...
          Edit a Protocol
          Menu Rebuild Menu ...
          Out-Of-Order Set Management ...
          See if a User Has Access to a Particular Option
          Show Users with a Selected primary Menu

Select Menu Management Option: KEY <Enter> Management

          Allocation of Security Keys
          De-allocation of Security Keys
          Enter/Edit of Security Keys
          All the Keys a User Needs
          Change user’s allocated keys to delegated keys
          Delegate keys
          Keys For a Given Menu Tree
          List users holding a certain key
          Remove delegated keys
          Show the keys of a particular user

Select Key Management Option: ALLOC <Enter> ation of Security Keys

Allocate key: XULM SYSTEM LOCKS

Another key: <Enter>

Holder of key: XUUSER,ONE <Enter>      OX          TECHNICAL WRITER

Another holder: <Enter>

You’ve selected the following keys:

XULM SYSTEM LOCKS

You’ve selected the following holders:

XUUSER,ONE

You are allocating keys.  Do you wish to proceed? YES// <Enter>

XULM SYSTEM LOCKS LOCKS being assigned to:
     XUUSER,ONE
```

14.3 Options

The **Lock Manager Menu** [XULM LOCK MANAGER MENU] is located on the **Operations Management** [XUSITEMGR] menu:

Figure 170: Lock Manager Menu [XULM LOCK MANAGER MENU]

```
Select Systems Manager Menu Option: OPER <Enter> ations Management

          System Status
          Introductory text edit
          CPU/Service/User/Device Stats
   LOCK   Lock Manager Menu ...
   RJD    Kill off a users’ job
          Alert Management ...
          Alpha/Beta Test Option Usage Menu ...
          Clean old Job Nodes in XUTL
          Delete Old (>14 d) Alerts
          Foundations Management
          Kernel Management Menu ...
          Post sign-in Text Edit
          User Management Menu ...

Select Operations Management Option: LOCK <Enter> Lock Manager Menu

   LM     Kernel Lock Manager
   EDIT   Edit Lock Dictionary
   LOG    View Lock Manager Log
   SITE   Edit Lock Manager Parameters
   PURG   Purge Lock Manager Log

Select Lock Manager Menu Option:
```

The **Lock Manager Menu** [XULM LOCK MANAGER MENU] includes the options listed in Table 19:

Table 19: Lock Manager—Options

Option Name	Option Menu Text	Description
XULM LOCK MANAGER	**Kernel Lock Manager**	Use this option to display the Lock Table and terminate processes that hold problem locks. This option is locked with the XULM LOCKS security key.
XULM EDIT LOCK DICTIONARY	**Edit Lock Dictionary**	User this option to add entries to the Lock Dictionary or edit existing entries.
XULM VIEW LOCK MANAGER LOG	**View Lock Manager Log**	Use this option to view the Kernel Lock Manager Log.

Option Name	Option Menu Text	Description
XULM EDIT PARAMETERS	**Edit Lock Manager Parameters**	Use this option to edit the site parameters for the Kernel Lock Manager.
XULM PURGE LOCK MANAGER LOG	**Purge Lock Manager Log**	Use this option to purge the Lock Manager Log of old entries.

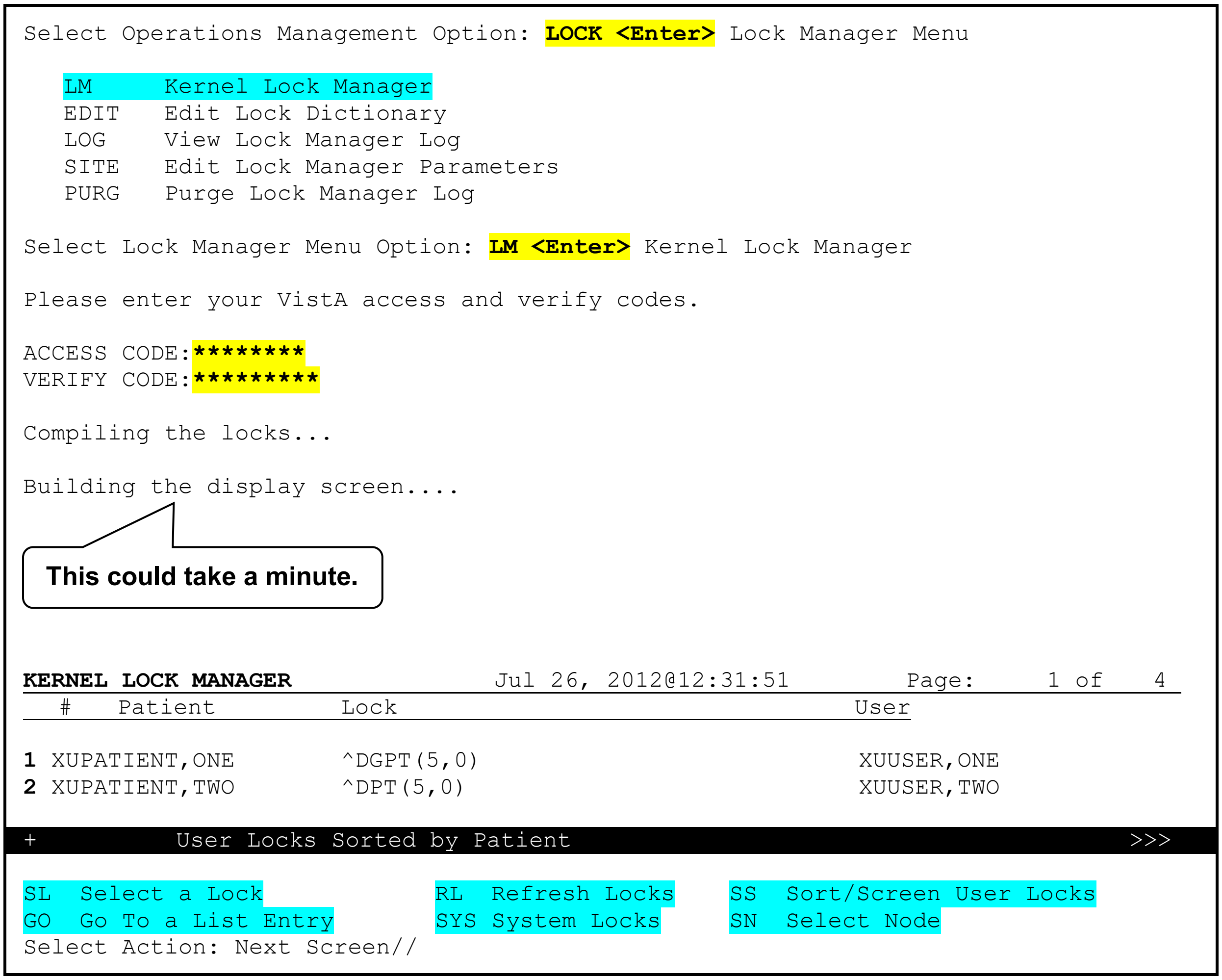
14.4 Using the Lock Manager

14.4.1 List Locks Screen

Use the **Kernel Lock Manager** [XULM LOCK MANAGER] option to view the lock table and the processes that own the locks. This option is locked with the XULM LOCKS security key.

Upon entering the option, you may be asked to enter your Access and Verify code. The Lock Manager uses these codes to query each node for information regarding locks and processes, via the RPC Data Broker. However, if the system consists of the single node on which you are already logged onto, you are *not* asked to enter your Access and Verify code.

Figure 171: Using Kernel Lock Manager Option [XULM LOCK MANAGER]—Sample User Entries and Report



```
Select Operations Management Option: LOCK <Enter> Lock Manager Menu

   LM     Kernel Lock Manager
   EDIT   Edit Lock Dictionary
   LOG    View Lock Manager Log
   SITE   Edit Lock Manager Parameters
   PURG   Purge Lock Manager Log

Select Lock Manager Menu Option: LM <Enter> Kernel Lock Manager

Please enter your VistA access and verify codes.

ACCESS CODE:********
VERIFY CODE:*********

Compiling the locks...

Building the display screen....
```

This could take a minute.

```
KERNEL LOCK MANAGER              Jul 26, 2012@12:31:51        Page:     1 of    4
   #   Patient         Lock                                    User

1 XUPATIENT,ONE        ^DGPT(5,0)                              XUUSER,ONE
2 XUPATIENT,TWO        ^DPT(5,0)                               XUUSER,TWO

+          User Locks Sorted by Patient                                        >>>

SL  Select a Lock            RL  Refresh Locks    SS  Sort/Screen User Locks
GO  Go To a List Entry       SYS System Locks     SN  Select Node
Select Action: Next Screen//
```

The main “User Locks” screen contains only user locks, as opposed to system locks. System locks are those locks used by infrastructure applications, such as the Kernel and HL7 packages, and are generally not of interest to users of the Lock Manager. In order to see the system locks, you can use the **SYS—System Locks** action.

Table 20 lists the actions available on the “**List Locks**” screen.

Table 20: Lock Manager—Actions

Lock Action	Description
SL—Select a Lock	This action allows a user to select a lock from the list. It then displays a new screen with detailed information about the lock.
GO—Go To a List Entry	This List Manager action asks the user where he/she wants to go to on the list and then shifts the display to that location.
RL—Refresh Locks	This action rebuilds the list of locks by reading the lock table.

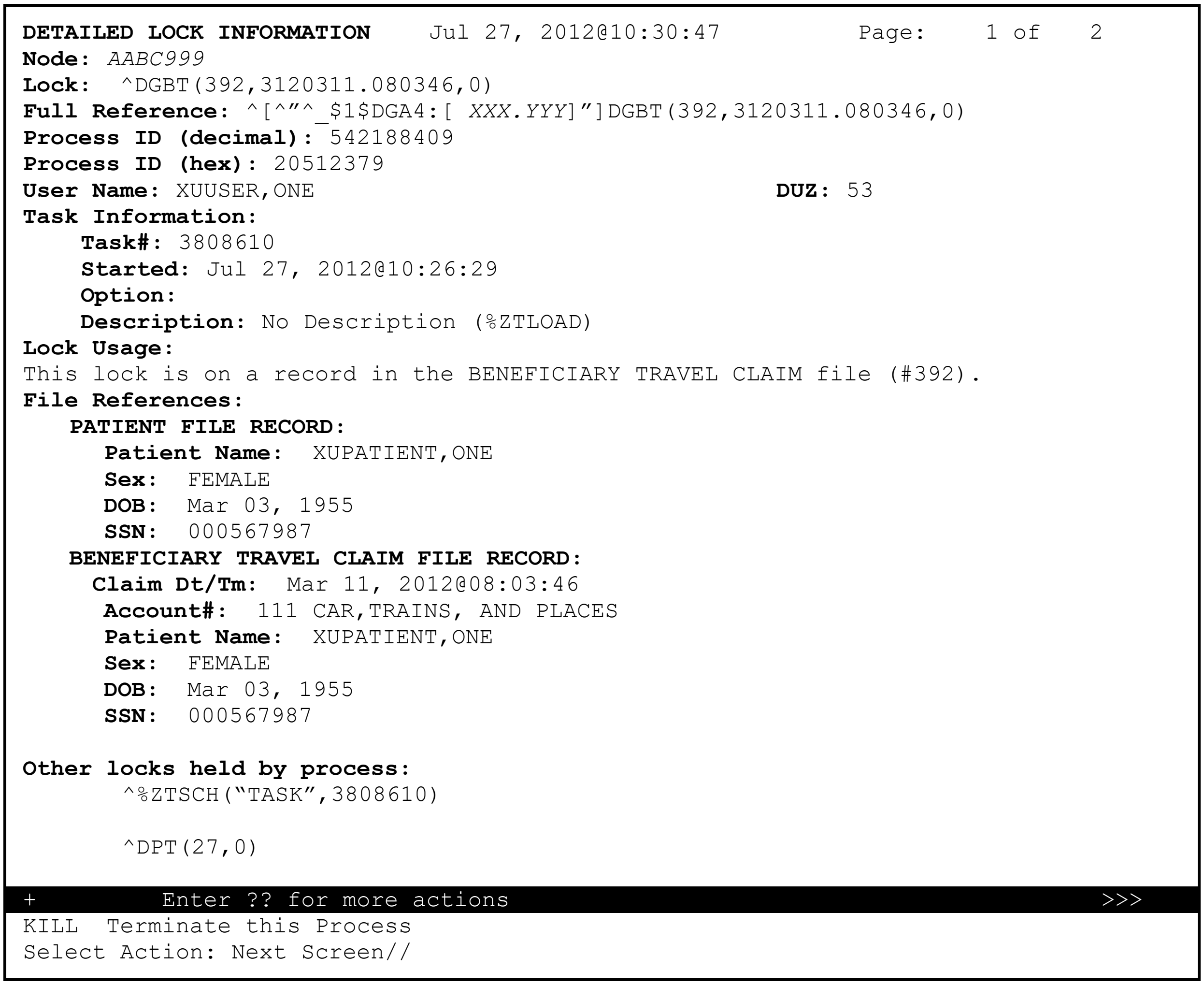
Lock Action	Description
SYS—System Locks	This action displays the list of the system locks. System locks are generally ignored within the Lock Manager. They are locks held by infrastructure packages, such as the Kernel or the HL7 package. ℹ **NOTE:** Only holders of the XULM SYSTEM LOCKS security key can use this option.
SS—Sort/Screen User Locks	This action provides the user with several options for how the list locks should be displayed. The options include sorting the list by the following: • Patient Name • User Name • Lock string, or screening the entries by lock reference, which means that only locks that relate to a specific file are included in the display.
SN—Select Node	This action allows the user to select either a single computer node or all computer nodes. If the user selects a single node, then the display of locks includes only locks placed by processes running on that node.

14.4.2 Single Lock Details Screen

Use the **SL—Select a Lock** action to view the lock details (Figure 172). The detailed information includes the following information:

- Node Information
- Lock ID
- Process ID (decimal and Hex)—Process that owns the lock.
- User Name
- Task Information
- Lock Usage
- File References—Files that the lock references
- Other locks held by process

Figure 172: Select a Lock Action—Sample Detailed Lock Information



```
DETAILED LOCK INFORMATION    Jul 27, 2012@10:30:47          Page:    1 of    2
Node: AABC999
Lock:  ^DGBT(392,3120311.080346,0)
Full Reference: ^[^”^_$1$DGA4:[ XXX.YYY]”]DGBT(392,3120311.080346,0)
Process ID (decimal): 542188409
Process ID (hex): 20512379
User Name: XUUSER,ONE                                 DUZ: 53
Task Information:
    Task#: 3808610
    Started: Jul 27, 2012@10:26:29
    Option:
    Description: No Description (%ZTLOAD)
Lock Usage:
This lock is on a record in the BENEFICIARY TRAVEL CLAIM file (#392).
File References:
   PATIENT FILE RECORD:
      Patient Name:  XUPATIENT,ONE
      Sex:  FEMALE
      DOB:  Mar 03, 1955
      SSN:  000567987
   BENEFICIARY TRAVEL CLAIM FILE RECORD:
     Claim Dt/Tm:  Mar 11, 2012@08:03:46
      Account#:  111 CAR,TRAINS, AND PLACES
      Patient Name:  XUPATIENT,ONE
      Sex:  FEMALE
      DOB:  Mar 03, 1955
      SSN:  000567987

Other locks held by process:
       ^%ZTSCH(“TASK”,3808610)

       ^DPT(27,0)

+         Enter ?? for more actions                                          >>>
KILL  Terminate this Process
Select Action: Next Screen//
```

14.4.2.1 Terminate this Process Action

Use the **KILL—Terminate this Process** action to terminate the process, thereby releasing all the locks held by it.

CAUTION: This action is irreversible! Before terminating a process, examine all the information provided on the screen. Do *not* terminate the process unless you are sure the user is no longer active.

Do *not* terminate a system process unless you have the expertise to ascertain the effect. Incorrectly terminating a system process could have adverse effects on multiple users or applications.

When a process is terminated, an entry is made in the XULM LOCK MANAGER LOG (#8993.2) file. It consists of the following data:

- User’s Name
- Date/Time of Action
- Detailed Lock Information

14.5 Managing the Lock Manager

Table 21 reviews the various management functions available within the Lock Manager and the corresponding option where the function can be performed.

Table 21: Lock Manager—Management Functions

Function	Option
Enable/Disable the Lock Manager	**Edit Lock Manager Parameters** [XULM EDIT PARAMETERS]
Edit IP address and port numbers of RPC Data Broker on the system nodes.	**Edit Lock Manager Parameters** [XULM EDIT PARAMETERS]
Edit the list of system locks. System locks are generally excluded from view within the Lock Manager, which makes it easier for users to review the lock table.	**Edit Lock Manager Parameters** [XULM EDIT PARAMETERS]
View the Lock Manager: use log that records each instance of a process being terminated.	**View Lock Manager Log** [XULM VIEW LOCK MANAGER LOG]
Purge the Lock Manager use log.	**Purge Lock Manager Log** [XULM PURGE LOCK MANAGER LOG]
Add or Edit entries in the Lock Dictionary.	**Edit Lock Dictionary** [XULM EDIT LOCK DICTIONARY]

14.6 Maintaining the Lock Dictionary

14.6.1 Adding Lock Templates—Edit Lock Dictionary Option

Use the **Edit Lock Dictionary** [XULM EDIT LOCK DICTIONARY] option to add to or edit entries in the XULM LOCK DICTIONARY (#8993) file.

A “Lock Template” is a description of the lock. It looks like the entry in the lock table, except that it can contain a variable in place of a subscript. A variable is used when the actual subscript value is *not* known in advance. Usually, it represents the internal entry number (IEN) of the record that is being locked. Variables are important, because they can be used in M code (see Figure 169).

To add an entry to the XULM LOCK DICTIONARY (#8993) file, perform the following procedure:

1. From the **Lock Manager Menu** [XULM LOCK MANAGER MENU] at the "Select Lock Manager Menu Option:" prompt, select the **Edit Lock Dictionary** [XULM EDIT LOCK DICTIONARY] option.
2. At the "Enter response: E//" prompt, enter one of the following values related to entries in the lock dictionary:
   - **E**—Edit an existing entry.
   - **D**—Delete an existing entry.
   - **A**—Add a new entry.

   In this example, the user is adding a new entry; so, she selected **A—Add a new entry**.
3. At the "LOCK TEMPLATE:" prompt, enter a lock template based on the following rules:
   - Locks are almost always on a global; though, it is allowable to lock a local variable. For the case of a global lock, enter a space as the first character, since VA FileMan does *not* allow a caret (^) as the first character (e.g., **^DGCR(399,IEN**; this sample includes a leading space before the ^).
   - Subscripts that are *not* variables should include quotes unless they are numbers.
   - Variables should start with a letter and should not be quoted.
4. At the "GLOBAL LOCK?: YES//" prompt, press **Enter** to accept the **YES** default. Locks are usually on globals, but it is possible to lock a local variable too.
5. At the "XULM LOCK DICTIONARY GLOBAL LOCK?: YES//" prompt, press **Enter** to accept the **YES** default.
6. At the "XULM LOCK DICTIONARY PACKAGE:" prompt, enter the package that is responsible for the lock (e.g., Integrated Billing [sample]).
7. At the "PARTIAL MATCH ALLOWED?:" prompt, enter **YES**. This means that a lock table entry with additional subscripts is still considered as matching the Lock template. For example, by answering **YES** to this prompt the lock on **^DGCR(399,1,0)** would be considered a match; otherwise, the additional subscript **0** would rule it out as a match.
8. At the "Edit? NO//" prompt, enter a description for the purpose of the Lock template.
9. (Optional) At the "Executable check logic for variable IEN (optional):" prompt, enter M code to verify that the variable IEN has a permissible value. It should set **Y=0** if *not* OK and **Y=1** if OK. For example:

   ```
   S Y=$S($D(^DGCR(399,IEN,0)):1,1:0)
   ```

   In this example, you can check that the record actually exists. If the check fails, then the Lock template is ruled *not* to match the lock. The M code should set **Y=1** if the value is acceptable, or **0** if the value is *not* acceptable. Setting **Y=0** means that the lock table entry is considered *not* to match the Lock template.

10. (Optional) At the “Select FILE:” prompt, you can enter a file that is related to the lock (e.g., PATIENT [#2] file) in some way. Either the lock is on a record in the file, or a record in the file can be navigated to based on the information contained within the lock.

    If you enter a file, then you can enter M code that returns identifiers from a record in that file that can help users identify the problem lock. If there are identifiers that you would like to display to the user, first select the file, and then enter the M code that retrieves the identifiers from the file.

    Users of the Lock Manager search for the problem lock by the file or files that it is related to. Entering “COMPUTABLE FILE REFERENCES” is what makes this possible. Most locks of interest are related in some way to a particular patient, so entries in the Lock Dictionary should almost always contain a computable file reference to the PATIENT (#2) file, but other computable file references should also be included when appropriate.

11. At the “Are you adding ‘*XXXXXXXX*’ as a new COMPUTABLE FILE REFERENCES (the *nXX* for this XULM LOCK DICTIONARY)? No//” prompt, enter **YES**.
12. At the “COMPUTABLE FILE REFERENCES FILE: *XXXXXXXX*//” prompt, press **Enter** to accept the default response.
13. At the “Enter MUMPS code that returns identifiers for the file:” prompt, enter M code that returns identifiers for the file references. In order to return identifiers for the PATIENT (#2) file, the application should call the PAT^XULMU API. It takes the patient DFN as the input. For example:

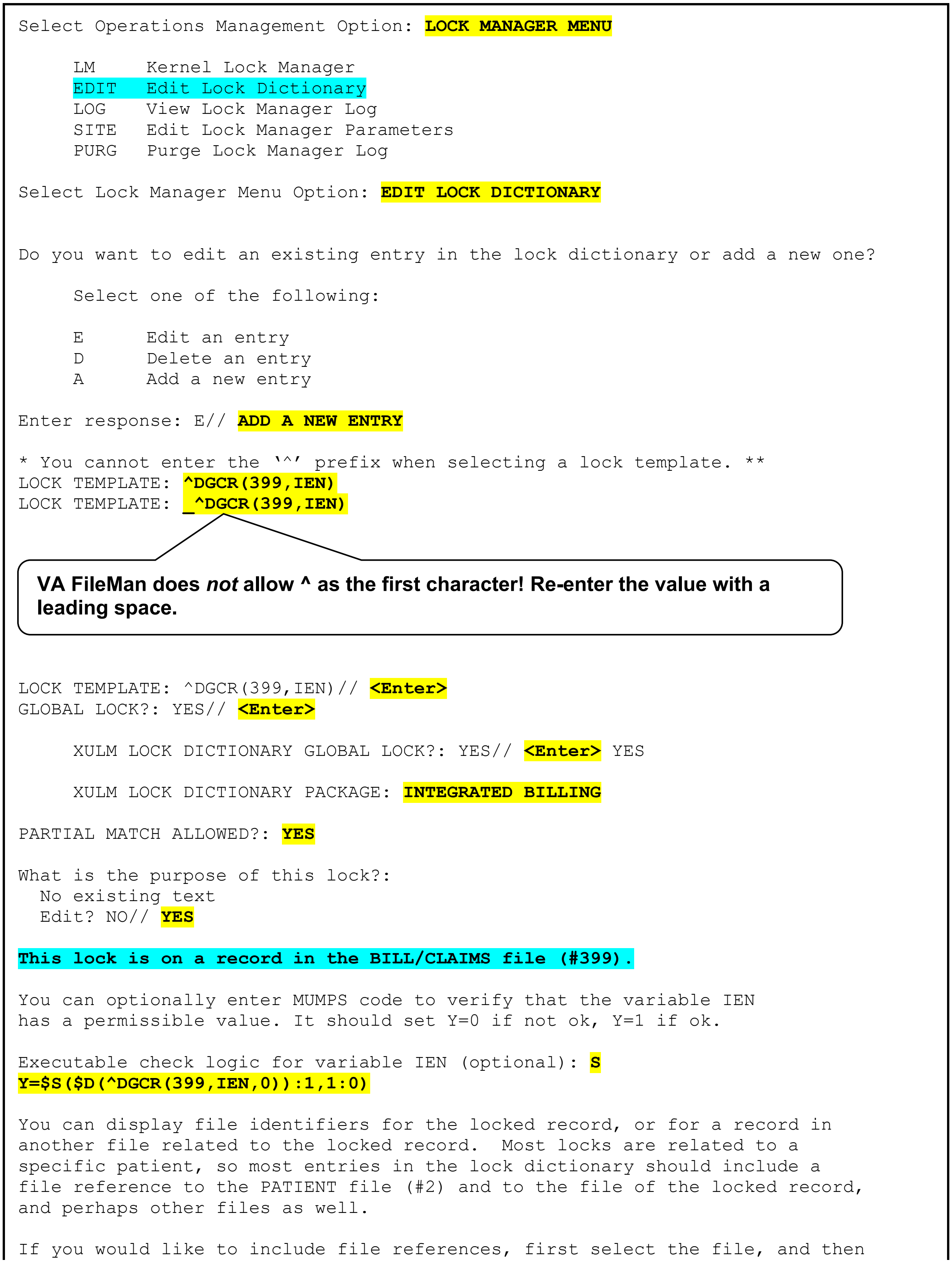
    ```
    D PAT^XULMU($P($G(^DGCR(399,IEN,0)),"^",2))
    ```

14. At the “Edit? NO//” prompt, enter **YES** and then enter a description to list the identifiers that are returned for this file reference (e.g., Name, Sex, Date of Birth [DOB], and Social Security Number [SSN]).
15. At the “Select FILE:” prompt, enter another computable file identifier (e.g., BILL/CLAIMS [#399] file).
16. At the “Are you adding ‘*XXXXXXXX*’ as a new COMPUTABLE FILE REFERENCES (the *nXX* for this XULM LOCK DICTIONARY)? No//” prompt, enter **YES**.
17. At the “COMPUTABLE FILE REFERENCES FILE: //” prompt, press **Enter**.
18. At the “Enter MUMPS code that returns identifiers for the file. MUMPS CODE:” prompt, enter M code that returns identifiers for the file references. This file returns identifiers from the PATIENT (#2) file as well as the bill number. In order to obtain the patient identifiers when the referenced file is *not* the PATIENT (#2) file use the ADDPAT^XULMU API. The input parameter is the patient DFN. For example:

    ```
    N ND S ND=$G(^DGCR(399,IEN,0)),ID(“IEN”)=IEN D ADDPAT^XULMU(+$P(ND,”^”,2)) S
    ID(0)=ID(0)+1,ID(ID(0))=”BILL NUMBER:”_$P(ND,”^”)
    ```

19. At the “Edit? NO//” prompt, enter **YES** and then enter a description to list the identifiers that are returned for this file reference (e.g., Name, Sex, Date of Birth [DOB], Social Security Number [SSN], and Bill Number).

Figure 173: Adding New Entry to XULM LOCK DICTIONARY (#8993) File—Sample ^DGCR(399,IEN) Template



```
Select Operations Management Option: LOCK MANAGER MENU

     LM     Kernel Lock Manager
     EDIT   Edit Lock Dictionary
     LOG    View Lock Manager Log
     SITE   Edit Lock Manager Parameters
     PURG   Purge Lock Manager Log

Select Lock Manager Menu Option: EDIT LOCK DICTIONARY


Do you want to edit an existing entry in the lock dictionary or add a new one?

     Select one of the following:

     E      Edit an entry
     D      Delete an entry
     A      Add a new entry

Enter response: E// ADD A NEW ENTRY

* You cannot enter the '^' prefix when selecting a lock template. **
LOCK TEMPLATE: ^DGCR(399,IEN)
LOCK TEMPLATE: _^DGCR(399,IEN)
```

VA FileMan does *not* allow ^ as the first character! Re-enter the value with a leading space.

```
LOCK TEMPLATE: ^DGCR(399,IEN)// <Enter>
GLOBAL LOCK?: YES// <Enter>

     XULM LOCK DICTIONARY GLOBAL LOCK?: YES// <Enter> YES

     XULM LOCK DICTIONARY PACKAGE: INTEGRATED BILLING

PARTIAL MATCH ALLOWED?: YES

What is the purpose of this lock?:
  No existing text
  Edit? NO// YES

This lock is on a record in the BILL/CLAIMS file (#399).

You can optionally enter MUMPS code to verify that the variable IEN
has a permissible value. It should set Y=0 if not ok, Y=1 if ok.

Executable check logic for variable IEN (optional): S
Y=$S($D(^DGCR(399,IEN,0)):1,1:0)

You can display file identifiers for the locked record, or for a record in
another file related to the locked record.  Most locks are related to a
specific patient, so most entries in the lock dictionary should include a
file reference to the PATIENT file (#2) and to the file of the locked record,
and perhaps other files as well.

If you would like to include file references, first select the file, and then
```

```
enter the MUMPS code that will retrieve the file identifiers from that file.

Select FILE: 2 <Enter> PATIENT
  Are you adding 'PATIENT' as
     a new COMPUTABLE FILE REFERENCES (the 1ST for this XULM LOCK DICTIONARY)? No
// YES
     COMPUTABLE FILE REFERENCES FILE: PATIENT// <Enter>

     Enter MUMPS code that returns identifiers for the file:

D PAT^XULMU($P($G(^DGCR(399,IEN,0)),"^",2))

     List the identifiers that are returned for this file reference.

     Identifiers:
     No existing text
     Edit? NO// YES

Returns the patient's name, sex, date of birth, and Social Security Number.

Select FILE: 399 <Enter> BILL/CLAIMS

     Are you adding 'BILL/CLAIMS ' as a new COMPUTABLE FILE REFERENCES (the 2ND for
this XULM LOCK DICTIONARY)? No// YES
     COMPUTABLE FILE REFERENCES FILE: // <Enter>

Enter MUMPS code that returns identifiers for the file.

     MUMPS CODE: N ND S ND=$G(^DGCR(399,IEN,0)),ID("IEN")=IEN D
ADDPAT^XULMU(+$P(ND,"^",2)) S ID(0)=ID(0)+1,ID(ID(0))="BILL NUMBER:"_$P(ND,"^")

List the identifiers that are returned for this file reference.

     Identifiers:
     No existing text
     Edit? NO// YES

This file reference returns the patient name, date of birth, sex,
Social Security Number, and BILL NUMBER.
```

14.6.2 Exporting Lock Templates

Entries in the Lock Dictionary can be included in a KIDS distribution. The KIDS enhancement that adds LOCK TEMPLATES as a new component are released in Kernel Patch XU*8.0*672.

14.7 Viewing and Purging Lock Manager Logs

14.7.1 View Lock Manager Log Option

Use the **View Lock Manager Log** [XULM VIEW LOCK MANAGER LOG] option to display the entries for the terminated lock processes in the XULM LOCK MANAGER LOG (#8993.2) file.

To view the Lock Manager log, perform the following procedure:

1. From the **Lock Manager Menu** [XULM LOCK MANAGER MENU], select the **View Lock Manager Log** [XULM VIEW LOCK MANAGER LOG] option.
2. At the “Select XULM LOCK MANAGER LOG DATE/TIME PROCESS TERMINATED:” prompt, enter a specific date/time or two question marks (**??**) to get a list.
3. .At the “DEVICE:” prompt, enter a device to display the log for the specific entry selected.

Figure 174: View Lock Manager Log Option [XULM VIEW LOCK MANAGER LOG]—Sample User Entries and Report

```
Select Lock Manager Menu Option: VIEW <Enter> Lock Manager Log
Kernel Lock Manager Log

Select XULM LOCK MANAGER LOG DATE/TIME PROCESS TERMINATED: ??

   Choose from:
   JUN 18, 2012@17:14:23
   JUN 18, 2012@17:22:32
   JUN 18, 2012@17:33:27
   JUN 19, 2012@09:03:58
   JUN 19, 2012@09:04:43
   JUN 19, 2012@09:45:49
   JUN 19, 2012@11:04:16
   JUN 19, 2012@11:06:47
   JUN 19, 2012@12:33:43
   JUN 19, 2012@12:35:36
   JUN 19, 2012@12:47:21
   JUN 19, 2012@12:48:48
   JUN 19, 2012@12:50:42
   JUN 19, 2012@12:53:16
   JUN 19, 2012@12:55:59
   JUN 20, 2012@06:40:46
   JUN 24, 2012@09:14:55
   JUN 24, 2012@09:21:43
   JUN 24, 2012@09:22:50
^
Select XULM LOCK MANAGER LOG DATE/TIME PROCESS TERMINATED: JUNE 18 <Enter> JUN 18,
2012
     1   6-18-2012@17:14:23
     2   6-18-2012@17:22:32
     3   6-18-2012@17:33:27
CHOOSE 1-3: 1 <Enter> 6-18-2012@17:14:23

DEVICE: <Enter>  Telnet Terminal    Right Margin: 80// <Enter>

XULM LOCK MANAGER LOG LIST                       AUG 14,2012  16:12    PAGE 1
--------------------------------------------------------------------------------

DATE/TIME PROCESS TERMINATED: JUN 18, 2012@17:14:23
  THE TERMINATOR: XUUSER,ONE
 PROCESS DESCRIPTION:
 Lock:  ^DGBT(1,0)
 Node: AABC999
 Full Reference: ^["^^_$1$DGA4:[NXT.NXT]”]DGBT(1,0)
 Process ID (decimal): 540943078
 Process ID (hex): 203E22E6
 User Name: UNKNOWN                                  DUZ:
 Task Information: not available
 Lock Usage:  not available
 File References: not available

 Other locks held by process:
        ^DGPT(1,0)

        ^DPT(4,0)

<Enter>

XULM LOCK MANAGER LOG LIST                       AUG 14,2012  16:12    PAGE 2
--------------------------------------------------------------------------------
```

```
         ^DPT(5,0)

Select XULM LOCK MANAGER LOG DATE/TIME PROCESS TERMINATED:
```

14.7.2 Purge Lock Manager Log Option

Use the **Purge Lock Manager Log** [XULM PURGE LOCK MANAGER LOG] option to purge the Lock Manager log.

To purge the Lock Manager log, perform the following procedure:

1. From the **Lock Manager Menu** [XULM LOCK MANAGER MENU], select the **Purge Lock Manager Log** [XULM PURGE LOCK MANAGER LOG] option.
2. At the “How many days of data should be retained: (0-365): 30//” prompt, enter the number of days to *retain* the log data (e.g., **30** days). Any log data older than the value entered is purged (e.g., **31** or more days). The default is **30** days with a maximum of **1** year (**365** days).
3. When the data purge is complete, the system displays: **DONE!**

Figure 175: Purge Lock Manager Log Option [XULM PURGE LOCK MANAGER LOG]—Sample User Entries and System Responses

```
Select Lock Manager Menu Option: PURG <Enter> Purge Lock Manager Log
How many days of data should be retained:  (0-365): 30// 364
DONE!
Enter RETURN to continue or '^' to exit:
```

14.8 Troubleshooting

14.8.1 Node Connection Error

If you get a node connection error (e.g., Figure 176) when using the **Kernel Lock Manager** [XULM LOCK MANAGER] option, contact your system administrator.

Figure 176: Sample Node Connection Error (Excerpt)

```
...

Compiling the locks...
Failed to connect to node 'ALTR4PA01': Connection error: Port, IP or server
logon error.
Continue with lock display? YES//
Failed to connect to node 'ALTR4PA02': Connection error: Port, IP or server
logon error.
Continue with lock display? YES//
Failed to connect to node 'ALTR4PA03': Connection error: Port, IP or server
logon error.
Continue with lock display? YES//

...
```

For example, instance names may *not* be defined in the Domain Name Service (DNS). System administrators can try pinging the name as it appears in the configuration, from one of the nodes in the configuration. Pinging should result in resolution to the IP address. If it does *not*, the instance name should be added to the DNS server that is used by these nodes.

III. Device Handler

15 Device Handler: User Interface

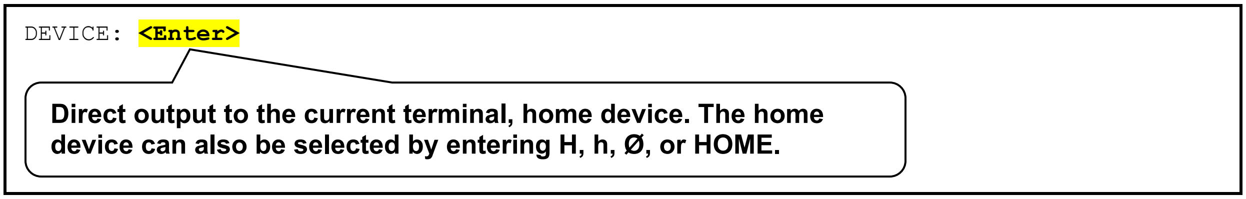
Applications that are designed for the Kernel environment perform output in a consistent manner, using Kernel's Device Handler. This ensures consistency on how you are asked to select devices for output and how the output is actually performed.

When you respond to the "DEVICE:" prompt, you are using the Device Handler.

15.1 Printing to Devices

At the "DEVICE:" prompt, to send output to your terminal, you can simply press the **<Enter>** key. This tells the Device Handler to display the report on the home device (that is, on your terminal), as shown in Figure 177:

Figure 177: Choosing the Home Device

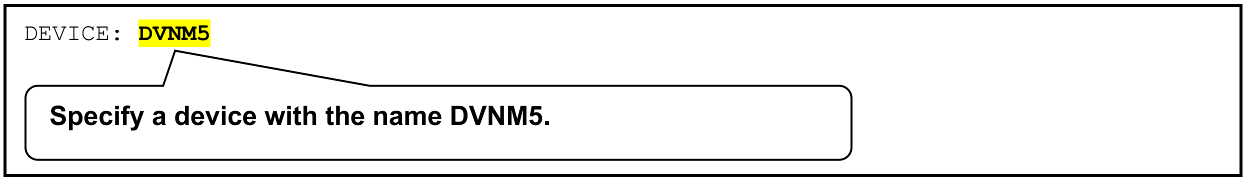


```
DEVICE: <Enter>
```

Direct output to the current terminal, home device. The home device can also be selected by entering H, h, Ø, or HOME.

To send output to a printer, enter the name of the printer at the "DEVICE:" prompt, as shown in Figure 178:

Figure 178: Choosing a Printer Device

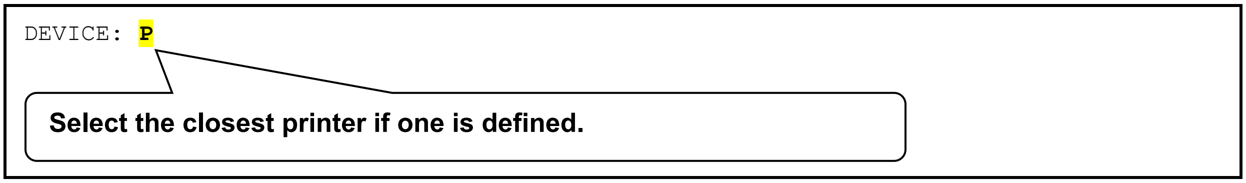


```
DEVICE: DVNM5
```

Specify a device with the name DVNM5.

To select the closest printer, if one is defined (unlikely), you can simply enter **P** and press **<Enter>**, as shown in Figure 179:

Figure 179: Choosing the Closest Printer Device

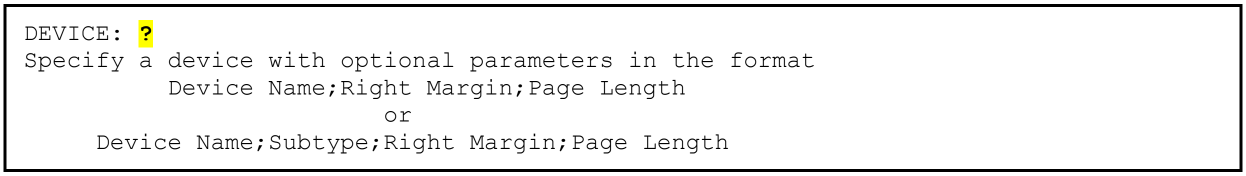


```
DEVICE: P
```

Select the closest printer if one is defined.

You can enter a question mark (**?**) to display help about the syntax of the response, as shown in Figure 180:

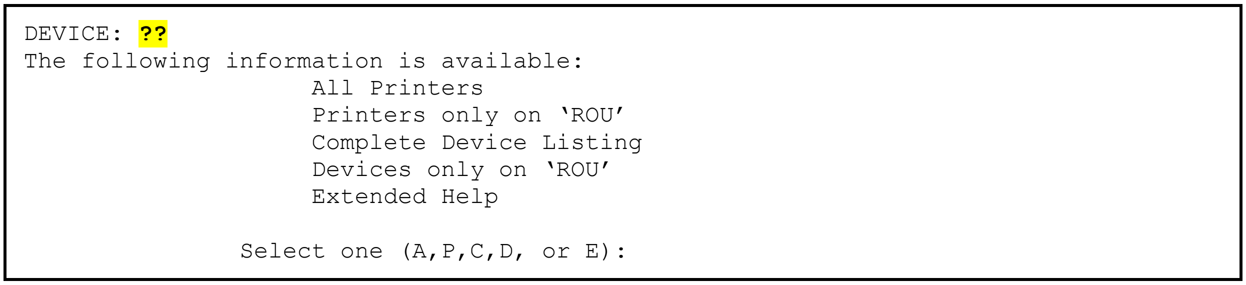
Figure 180: Device Syntax Help—One Question Mark (?)



```
DEVICE: ?
Specify a device with optional parameters in the format
          Device Name;Right Margin;Page Length
                         or
     Device Name;Subtype;Right Margin;Page Length
```

You can enter two question marks (**??**) to display available printers and other devices connected to the current Volume Set or "reachable from" the current Volume Set. You can also ask for a series of help frames under extended help, as shown in Figure 181:

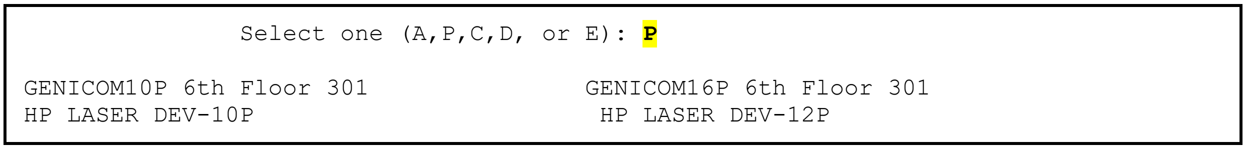
Figure 181: Displaying Devices Help—Two Question Marks (??)



```
DEVICE: ??
The following information is available:
                    All Printers
                    Printers only on 'ROU'
                    Complete Device Listing
                    Devices only on 'ROU'
                    Extended Help

               Select one (A,P,C,D, or E):
```

You can list all devices. In addition to printers, this list shows other types of devices you can use to handle output. An example of a partial printer listing is shown in Figure 182:

Figure 182: Sample Printer Listing



```
               Select one (A,P,C,D, or E): P

GENICOM10P 6th Floor 301               GENICOM16P 6th Floor 301
HP LASER DEV-10P                        HP LASER DEV-12P
```

REF: Unusual device types (e.g., Resource devices) are discussed in "Special Devices."

15.1.1 Specifying Right Margin and Page Length

Ordinarily, when choosing an output device, you only need to specify the device name. There can be times, however, when you may find it useful to specify the right margin or the page length for your output. The syntax to specify margin and page length uses semicolon delimiters. The format is:

```
DEVICE: Device Name ; Right Margin ; Page Length
```

The examples in Table 22 show how to use the additional semicolon-delimited pieces at the "DEVICE:" prompt:

Table 22: Sample Semicolon-delimited Pieces at the "DEVICE:" Prompt

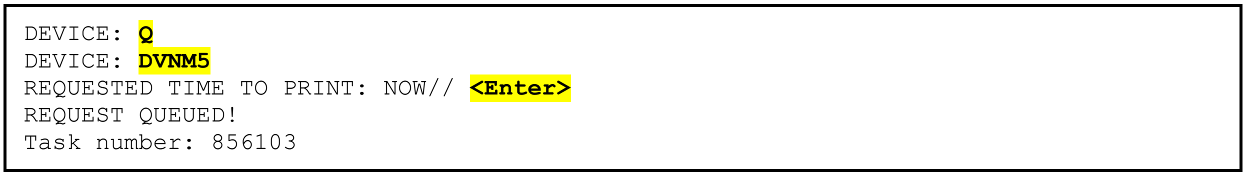
Semicolon-delimited Piece	Description
DEVICE: **DVNM5;80;66**	Use the DVNM5 device with a right margin of 80 columns and page length of 66 lines.
DEVICE: **;132**	Use the home device, right margin of 132.
DEVICE: **;;66**	Use the home device and format the output with page breaks at 66 lines.
DEVICE: **;;9999**	Scroll output on the home device without needing to press the **<Enter>** key at page breaks.

15.2 Queuing

At the "DEVICE:" prompt, if you enter a device's name, the output goes directly to that device. If the output you're sending is, for example, a long report, this ties your terminal up until the report finishes printing to that device.

You can print output and yet keep your terminal free for other processing by queuing your jobs rather than running them directly. As described in the "TaskMan: User Interface" section, you can queue output by entering **Q** at the "Device:" prompt. The device prompt is then presented a second time so that you can specify the output device.

Figure 183: Specifying a Device and Queuing a Print Job—Sample User Dialog (1 of 2)



```
DEVICE: Q
DEVICE: DVNM5
REQUESTED TIME TO PRINT: NOW// <Enter>
REQUEST QUEUED!
Task number: 856103
```

Alternatively, you can still specify the device first. The Device Handler checks to see if the device is available and, if so, asks you if you want to queue your output. If the device *cannot* be reached at the current time, Device Handler indicates that the device is busy or unavailable. You can avoid the preliminary availability check by entering **Q** at the first prompt (see Figure 183).

Figure 184: Specifying a Device and Queuing a Print Job—Sample User Dialog (2 of 2)

```
DEVICE: DVNM5
DO YOU WANT YOUR OUTPUT QUEUED? NO// YES

REQUESTED TIME TO PRINT: NOW// T@18:00 <Enter>  (JUL 11, 2004@18:00)
REQUEST QUEUED!
Task number: 856109
```

Whether you request queuing before or after naming a device, Device Handler then asks you to specify a time for the queued job to run. You can accept the default (NOW) or indicate a later time in the usual format. Queuing sends output to TaskMan for scheduling. Meanwhile, you can continue working on the computer system without a delay.

Figure 185: Queuing a Print Job—Sample User Dialog

```
REQUESTED TIME TO PRINT: NOW// T@18:00 <Enter>  (JUL 11, 2004@18:00)
REQUEST QUEUED!
Task number: 856109
```

REF: For more information about queuing output, see the "TaskMan: User Interface" section.

15.3 Specifying a Special Subtype

There is an exception to using numbers in the second semicolon piece to indicate a right margin setting. If, instead of a number, you use a letter and then a hyphen in a device specification (e.g., P-DEC), the second semicolon piece specifies a terminal type entry from the TERMINAL TYPE (#3.2) file to use for the output. A terminal type entry specifies information about what commands to use with specific printers (e.g., escape codes).

Figure 186: Terminal-Type Device Entry—*Without* Pauses

```
DEVICE: ;P-DEC
```

If the home device is a video terminal, output would be formatted with page breaks, and it would scroll without waiting for the user to press the <Enter> key after a screen display.

One form of the subtype request made possible by VA FileMan's print routines is the use of the word SINGLE along with **P-** or **PK-**. Appending **-SINGLE** indicates that a pause should occur after the display of each page. If using a slaved device to print the screen display, for example, the next page is displayed only after the user has pressed **<Enter>**:

Figure 187: Terminal-Type Device Entry—*With* Pauses

```
DEVICE: ;P-DEC-SINGLE
```

If the home device is a video terminal, output would be presented one (single) page at a time; the next page being displayed after the user presses the <Enter> key.

If you are *not* sure which subtype to use, you can enter a partial specification of the subtype in the second piece, and the Device Handler lets you choose from all matching subtypes. For example, if a dozen subtypes begin with "**P-LASER**...," you can list them by entering only the beginning of the subtype name (e.g., **P-LASER**):

Figure 188: Partial Device Specification—Unknown Subtype

```
DEVICE: LASER;P-LASER
```

All subtypes beginning with **P-LASER** are listed; you can then choose a subtype from this list.

When using a subtype as the second semicolon piece of a device specification, you can still specify a right margin and page length to use, but you then do so with the 3rd and 4th semicolon pieces:

Figure 189: Device Specification—Four-Semicolon Piece: Sample

```
DEVICE: LASER;P-LASER-NEW;132;100
```

The syntax for the four-semicolon piece form of the device specification is:

Figure 190: Device Specification—Four-Semicolon Piece: Syntax

```
DEVICE: Device Name ; Subtype ; Right Margin ; Page Length
```

15.3.1 Spool Document Names—An Exception

When you request the spool device at the device prompt, you can use the formats in Figure 191 and Figure 192 to specify the spool document name:

Figure 191: Device Syntax—Specifying a Spool Document Name: Sample Formats (1 of 2)

```
DEVICE: Spooler ; Spool Document Name ; Right Margin ; Page Length
```

Figure 192: Device Syntax—Specifying a Spool Document Name: Sample Formats (2 of 2)

```
DEVICE: Spooler ; Subtype ; Spool Document Name
```

Although neither right margin nor page length can be specified when including a subtype as the second piece and spool document name as the third, no functionality is lost. The explanation is simple; the spooler only responds to these two terminal type specifications. In other words, identifying a subtype for the spooler does no more than define a margin and page length.

Spool document entries in the SPOOL DOCUMENT (#3.51) file *cannot* have names beginning with: **P-**, **PK-**, **C-**, etc. (i.e., one or two letters followed by a hyphen, see Section 16.4.1). Because this syntax is the required naming convention for subtypes, you are allowed to specify the document name and the subtype in any order.

REF: For more information about Spool Devices, see the "Spooling" section.

15.4 Alternate Syntax for Device Specification

An alternate syntax is available for specifying right margin and page length when responding to the device prompt. Using the alternate format, you can specify pitch, intensity, and quality. The success of specifying these additional attributes, however, depends on whether the corresponding fields have been defined by system administrators at your site.

The syntax requires the use of a slash ("/") after the last semicolon (see Figure 193).

You can use the codes in Table 23 to specify special device attributes (in any order), without separating punctuation to delimit the pieces:

Table 23: Alternate Device Attribute Codes

Code	Description
B	Boldface
L	Page length
M	Margin
P	Pitch
Q	Quality (can be Q, Q1, or Q2)

For example, you could specify:

Figure 193: Specifying a Device—Using Alternate Syntax

```
DEVICE: LASER;P-LASER-LANDSCAPE;/M132L100P16BQ2
```

In this example (Figure 189), the following attributes are set:

- Margin ("**M**") is set to 132 (**M132**)
- Page length ("**L**") is set to 100 lines (**L100**)
- Pitch ("**P**") is set to 16 (**P16**)
- Intensity to boldface ("**B**")
- Quality ("**Q**") set to letter quality (**Q2**).

An absence of the **B** would indicate normal intensity. The quality settings are: **Q**, **Q1**, and **Q2**.

Your system administrators need to confirm that the appropriate code to set the specified printer attributes is set up for the device that you are using. Then, when the Device Handler closes the device, system administrators need to be sure that appropriate reset code is in the CLOSE EXECUTE field, so that the characteristics do *not* stay in effect. If, for example, someone requests a small pitch, subsequent reports also use the small pitch unless reset in the CLOSE EXECUTE statement for that device (or altered by the OPEN EXECUTE statement of the next device called).

15.5 Summary

The Device Handler is a common interface used by all VistA applications to send output to devices (usually, printers). Once you become familiar with the Device Handler, you can enhance your productivity by making use of some of the Device Handler's special features, including queuing, selecting a specific right margin or page length, and selecting a special subtype.

16 Device Handler: System Management

The Device Handler makes use of two primary Files:

- DEVICE (#3.5) File
- TERMINAL TYPE (#3.2) File

Together, these two files control most of the characteristics of devices in Kernel.

The global locations of the device-related files are listed in Table 24:

Table 24: Device-related Files Global Locations

Device-related File Name	Global Location
DEVICE (#3.5)	**^%ZIS(1,**
TERMINAL TYPE (#3.2)	**^%ZIS(2,**
DA RETURN CODES (#3.22)	**^%ZIS(3.22,**

16.1 DEVICE (#3.5) File

Kernel's DEVICE (#3.5) file stores information about devices on the system. All connected volume Sets/CPUs should make use of a single DEVICE (#3.5) file. Then all information concerning a particular device is stored in just one place, which facilitates device management.

Sometimes, a CPU has an attachment point to which a device can be connected, for example, physical ports. The **$I** field in the DEVICE (#3.5) file entry identifies this attachment point.

Most devices (e.g., printers) are connected to the network and **$I** points to the name used by the underlying OS to point to the device. When using such a device, Kernel's Device Handler allows the creation and use of multiple DEVICE (#3.5) file entries for the same physical device. Each DEVICE (#3.5) file entry can contain different specifications (font, margin, page length, etc.) to format output. Each entry in the DEVICE (#3.5) file, then, uniquely identifies a set of instructions to send to a particular device on the network.

Each device that Kernel Device Handler needs to communicate with should be set up as an entry in the DEVICE (#3.5) file. The DEVICE (#3.5) file supports a variety of devices, including video display terminals (VDTs), commonly called cathode ray tube devices (CRTs); printers; tape drives; and operating system files (e.g., Host File Server [HFS] devices).

The DEVICE (#3.5) file is located in the Manager's account for common reference from all associated accounts. With TaskMan's help, this information is also available to all associated processors (CPUs) in the local area network.

16.1.1 DEVICE File Fields

The most essential fields in the DEVICE (#3.5) file to populate or consider populating for device entries are listed in Table 25:

Table 25: DEVICE File Fields

Field	Description
NAME (#.01)	This is the name of the device. It is used at the “DEVICE” prompt to select this device. It should *not* be the internal name for the device but a logical one. It *must* start with one uppercase character and *not* contain lowercase characters.
$I (#1)	This field holds the hardware port name that the operating system (OS) can identify when referencing a port on a CPU. On layered systems where opening of host files is supported, this field can hold the host file name.
VOLUME SET(CPU) (#1.9)	(Optional) This field holds the name of the CPU to which this device belongs. It holds the name of the CPU where the physical port resides. If entered, the device is assumed to be accessible only from the specified CPU. If the field is left blank, this device is assumed to be accessible from all CPUs in the network. In other words, when this device is referenced, the Device Handler operates as if this device is resident on the local CPU. For example, if there is a device that uses the same **$I** on each CPU, one entry can be made in the DEVICE (#3.5) file by leaving this field set to **NULL**. This shortcut works only if the same **$I** has been associated with this device on every CPU. The Device Handler still maintains the CPU cross-reference to support queuing and other activities. The cross-reference format involves use of periods as delimiters. If the VOLUME SET(CPU) value were “**BBB**,” the cross-reference for the device with a **$I** of **75** would be “**BBB.75**”. If the VOLUME SET(CPU) value were **NULL**, then “**.75**” would be the CPU cross-reference. **NOTE:** In the Caché environment, where cluster mounting is used and most devices are set up on all CPUs, all such devices do *not* need a value for this field.
SIGN-ON/SYSTEM DEVICE (#1.95)	If set to **YES**, this field identifies that this entry is the primary device among those device entries that have the same **$I** with the same VOLUME SET(CPU) . Among those device entries that have a common **$I** and CPU, only one of these entries can have this field set to **YES**. If none of the common

Field	Description
	device entries is set to **YES**, the default device is identified by the first device on the CPU cross-reference. The default device is used when the Device Handler is invoked with **$I** as the device to be selected.
TYPE (#2)	This field contains the general type of device on the CPU. ℹ **REF:** For a list of device types, see Table 26.
SUBTYPE (#3)	Use this field to select a default terminal type for the device. This field points to the TERMINAL TYPE (#3.2) file to retrieve a standard set of characteristics that have been defined for vendor devices (e.g., Laser printers or VT320 CRTs). ℹ **REF:** For a discussion of the TERMINAL TYPE (#3.2) file, see the "TERMINAL TYPE (#3.2) File" section.
QUEUING (#5.5)	You can control the degree of queuing allowed for a device with the QUEUING field. ℹ **REF:** For a list of settings to control queuing for a device, see Table 27.
PRE-OPEN EXECUTE (#7)	This is the executable M code that is used by **%ZIS** before opening the device. If you define the **%ZISQUIT** variable, the device open fails. Setting **%ZISQUIT=1** in the PRE-OPEN EXECUTE code signals **%ZIS** to reject the selected device. With this variable, you can use the PRE-OPEN EXECUTE as a screen on whether the device should be opened or not.
POST-CLOSE EXECUTE (#8)	This is the executable M code that is used by **%ZISC** after closing the device.
OPEN PARAMETERS (#19)	These parameters are used to open a device with specified characteristics/addresses. This field is primarily used with non-terminal devices (e.g., Magtape and Sequential Disk Processor). Magtape (MT), SDP (obsolete), and Host File Server (HFS) device types use the value in this field as the default if the ASK PARAMETERS (#5) flag is set. Users would then be prompted for address/parameters. If the ASK PARAMETERS flag is *not* set and if there is a value in the OPEN PARAMETERS (#19) field, this value is used when opening the device (or file).

Field	Description
	NOTE: Each operating system has its own way of specifying parameters. For example, under Caché, margins are set with both the **OPEN** and **USE** command.
USE PARAMETERS (#19.5)	This field holds the parameters to be used in an M **USE** statement. The Device Handler takes information from this field when opening and using such devices as the Magtape (MT) drive. **NOTE:** Each operating system has its own way of specifying parameters. For example, under Caché, margins are set with both the **OPEN** and **USE** command.

Table 26: Device Types in the TYPE Field in the DEVICE (#3.5) File

Type	Description
BAR	Identifies the device as a bar code reader.
CHAN	Network Channels are high speed devices that use network protocols (e.g., TCP/IP).
HFS	The Host File Server (HFS) type and the associated functionality provides the vehicle to **READ** and **WRITE** to host level files. Instead of directing reports to a printer, the results could be placed into an OpenVMS or UNIX/Linux file. This would allow non-M-based statistical software or spreadsheet to use data produced by the M-based application by simply extracting data from the host file.
IMPC	Imaging work station device (reserved for VistA Imaging).
MT	Magtape (MT) devices.
OTH	Other (OTH) devices that do *not* fit a particular category.
RES	Resources (RES) is a type used for special sequencing of tasks that do *not* require a particular device.
SDP	(Obsolete) Sequential Disk Processor (SDP) is a predefined allocated disk space used for sequential processing; use HFS.
SPL	Spool (SPL) device is a predefined allocated disk space. It is similar to SDP; however, access to the spool device can be achieved from multiple users simultaneously.
TRM	Terminal devices (e.g., most CRTs and printers) should be associated with a corresponding device entry with a type of **TRM**.
VTRM	Virtual Terminal Server devices are those that are associated with a dynamically created M port identification (**$I**). A generic device entry with a device type of

Type	Description
	VTRM can be established for users who log into the system through terminal servers or other network protocols.

NOTE: Device type descriptions can also be obtained by entering two question marks (**??**) at the TYPE field while editing a device.

REF: For more information on these device types, see "Special Devices."

Also, for more information on Host File Server (HFS) devices, see "Host Files.

Table 27: Queuing Settings

Setting	Queuing	Description
0	ALLOWED	Jobs can be queued or run directly (default).
1	FORCED	Queuing is forced, unless disallowed by application.
2	NOT ALLOWED	Queuing to device is *not* allowed.

16.1.1.1 OpenVMS-Specific DEVICE Fields

NOTE: These fields are used by VMS and *not* Caché.

The DEVICE (#3.5) file can store operating system-specific information. For example, several fields are included in the DEVICE (#3.5) file to configure terminals and ports on Terminal Servers as part of an OpenVMS start-up command file. These operating system-specific fields in the DEVICE (#3.5) file are listed in Table 28:

Table 28: Mixed OS Environment Fields in the DEVICE (#3.5) File

Field	Description
LAT SERVER NODE (#61)	This is the DECserver/terminal server node name that the device is on. It is used by the **XTLATSET** routine to build data files for VMS startup.
LAT SERVER PORT (#62)	This is the port on the DECserver/terminal server to which this device is connected. It can be entered in the **LC-2-5** form or **31** form. On

Field	Description
	EQUINOX it is in the **PORT_31** form. This field is used by the **XTLATSET** routine to build VMS data files for startup.
VMS DEVICE TYPE (#63)	This is a flag that is passed into the file LT_PTR.DAT by the **XTLATSE**T routine to select how this port should be set up in VMS by the **SYS$MANAGER:SYSPRINT.COM** file when it runs.
LAT PORT SPEED (#64)	This field holds the value that is passed to the TSC_LOAD.COM file for loading the DECserver permanent database.
PRINT SERVER NAME OR ADDRESS (#65)	This field contains the fully qualified domain name (FQDN) or specific TCP/IP address of a remote server (e.g., LPD/LPR printing) or device (e.g., Telnet printer).
TELNET PORT (#66)	This field contains the Telnet port of a remote device (e.g., Telnet printer). The allowable range is a number between **2000** and **65534**.
REMOTE PRINTER NAME (#67)	This is the name of the remote printer that is referenced by the PRINT SERVER NAME OR ADDRESS (#65) and TELNET PORT (#66) fields.

Kernel Toolkit software distributes the **XTLATSET** and **NVSTNSET** routines that makes use of these fields.

16.1.2 Device Edit Menu

The DEVICE (#3.5) file has many more fields where additional specific information for particular devices can be entered. Kernel provides a number of options to facilitate creating and editing device types on the **Device Edit** [XUDEVEDIT] menu, which is located on the **Device Management** [XUTIO] menu:

Figure 194: Device Edit Options

```
Device Management ...                                                [XUTIO]
  Device Edit                                                    [XUDEVEDIT]
    ALL          Edit All Device Fields                       [XUDEVEDITALL]
    CHAN         Network Channel Device Edit                 [XUDEVEDITCHAN]
    HFS          Host File Server Device Edit                 [XUDEVEDITHFS]
    LPD          LPD/VMS Device Edit                          [XUDEVEDITLPD]
    MT           Magtape Device Edit                           [XUDEVEDITMT]
    RES          Resource Device Edit                         [XUDEVEDITRES]
    SPL          Spool Device Edit                            [XUDEVEDITSPL]
    TRM          TRM or VTRM Device Edit                      [XUDEVEDITTRM]
```

AUTHOR'S NOTE: The **SDP Device Edit** [XUDEVEDITSDP] option is purposely *not* displayed in this menu list, because it is obsolete.

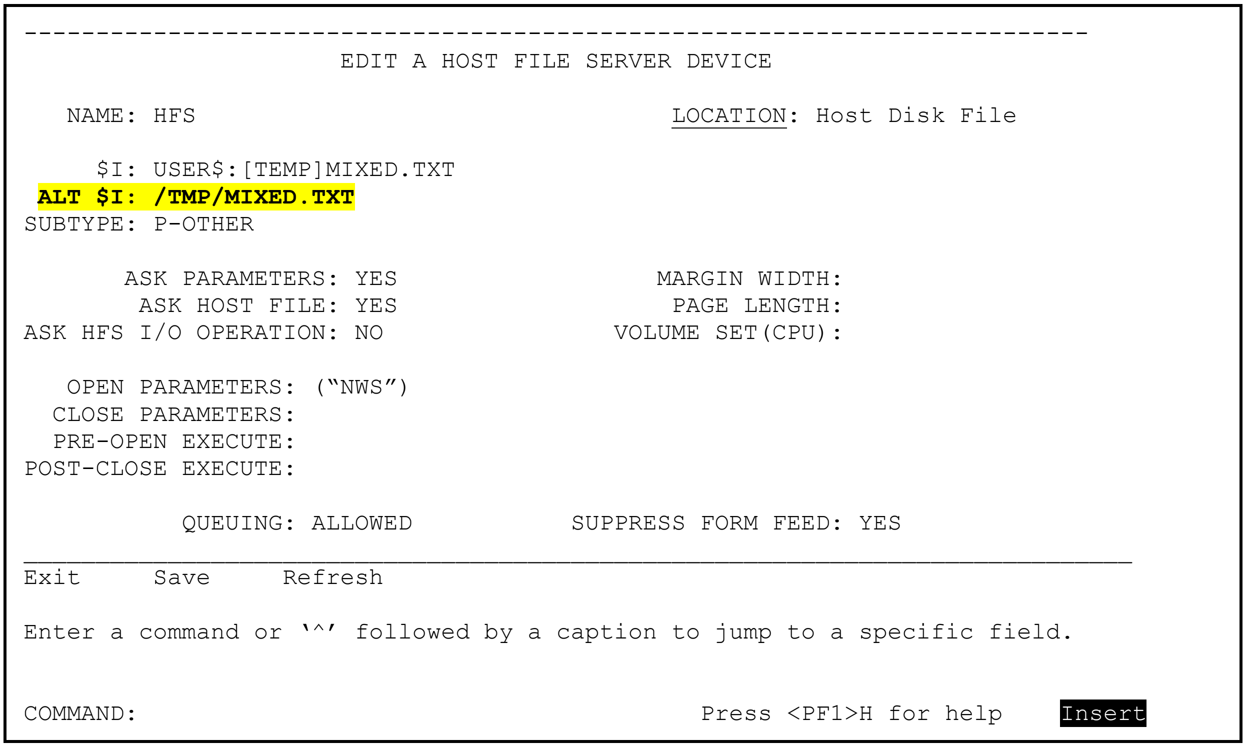
16.1.3 Sample Device File Entries

Kernel patch XU*8.0*440 also included the addition of the SECONDARY $I (#52) field in the DEVICE (#3.5) file.

16.1.3.1 HFS Devices

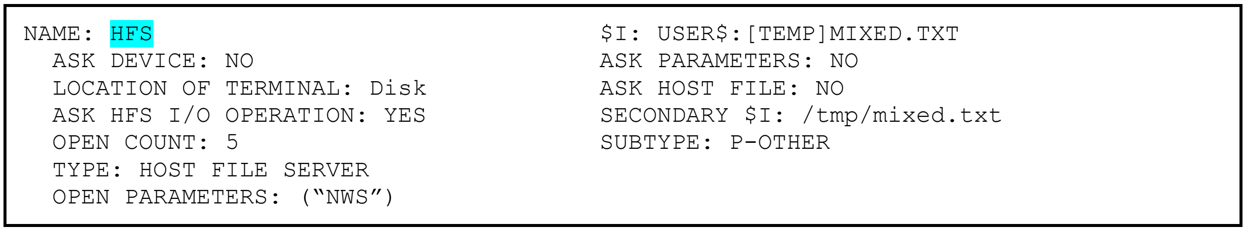
Figure 195 and Figure 196 show an HFS device using the **Host File Server Device Edit** [XUDEVEDITHFS] option to update the SECONDARY $I (#52) field in the DEVICE (#3.5) file:

Figure 195: HFS Device—Sample Data Entry Screen



```
--------------------------------------------------------------------------
                     EDIT A HOST FILE SERVER DEVICE

   NAME: HFS                                LOCATION: Host Disk File

     $I: USER$:[TEMP]MIXED.TXT
 ALT $I: /TMP/MIXED.TXT
SUBTYPE: P-OTHER

       ASK PARAMETERS: YES                   MARGIN WIDTH:
        ASK HOST FILE: YES                    PAGE LENGTH:
ASK HFS I/O OPERATION: NO                  VOLUME SET(CPU):

   OPEN PARAMETERS: ("NWS")
  CLOSE PARAMETERS:
  PRE-OPEN EXECUTE:
POST-CLOSE EXECUTE:

           QUEUING: ALLOWED           SUPPRESS FORM FEED: YES
_____________________________________________________________________________
Exit     Save     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                      Press <PF1>H for help    Insert
```

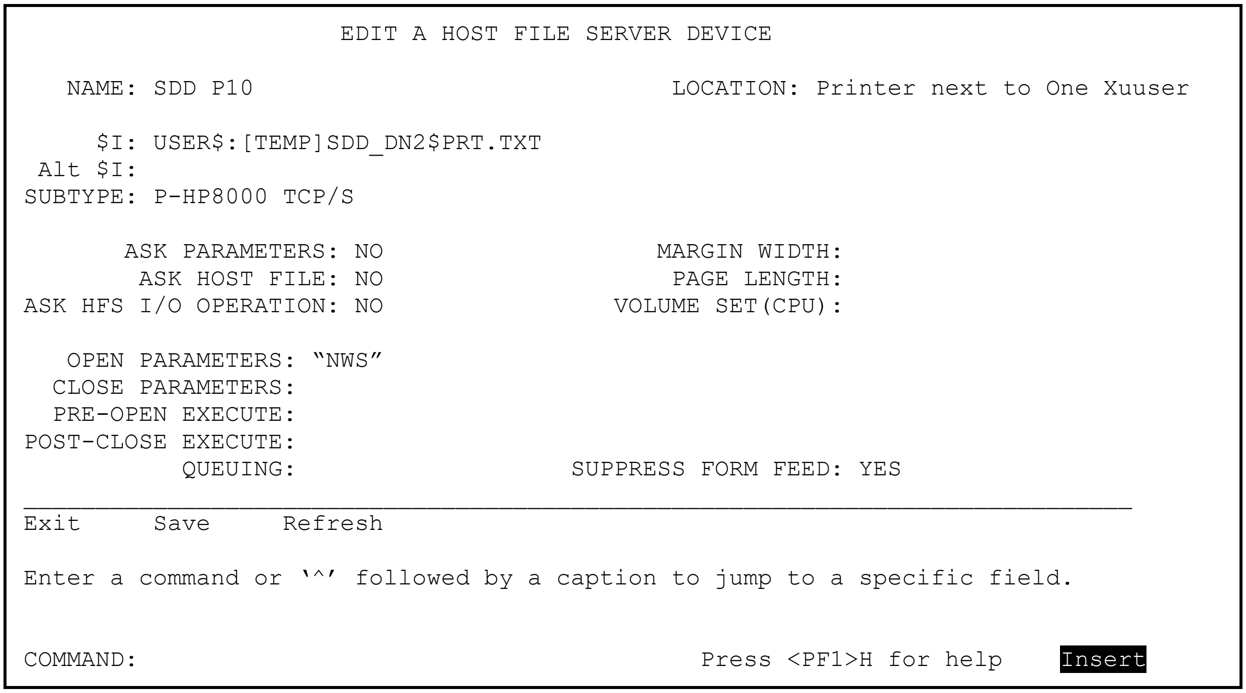
Figure 196: HFS Device—Sample DEVICE File Entry



```
NAME: HFS                                   $I: USER$:[TEMP]MIXED.TXT
  ASK DEVICE: NO                            ASK PARAMETERS: NO
  LOCATION OF TERMINAL: Disk                ASK HOST FILE: NO
  ASK HFS I/O OPERATION: YES                SECONDARY $I: /tmp/mixed.txt
  OPEN COUNT: 5                             SUBTYPE: P-OTHER
  TYPE: HOST FILE SERVER
  OPEN PARAMETERS: ("NWS")
```

Figure 197 shows a printer set up as an HFS device with the Terminal Type CLOSE EXECUTE, which submits the file to the OS print queue:

Figure 197: HFS Device—Sample Data Entry Screen with the Terminal Type CLOSE EXECUTE

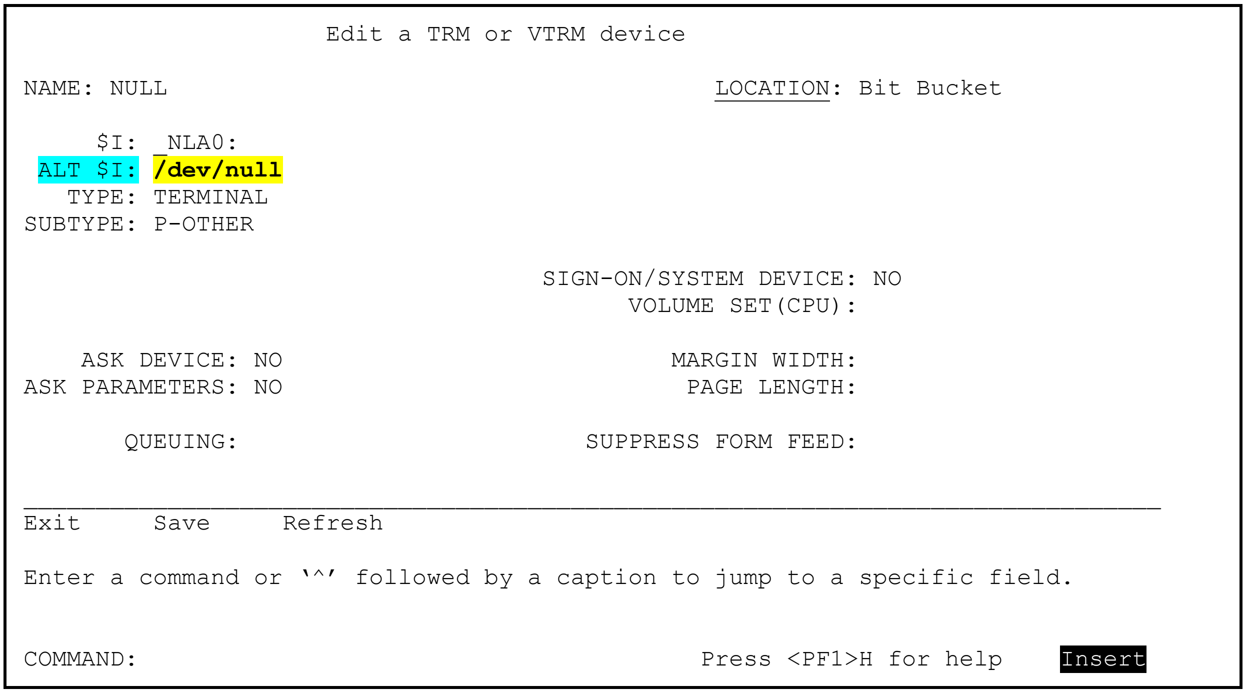


```
                      EDIT A HOST FILE SERVER DEVICE

   NAME: SDD P10                             LOCATION: Printer next to One Xuuser

     $I: USER$:[TEMP]SDD_DN2$PRT.TXT
 Alt $I:
SUBTYPE: P-HP8000 TCP/S

       ASK PARAMETERS: NO                        MARGIN WIDTH:
        ASK HOST FILE: NO                         PAGE LENGTH:
ASK HFS I/O OPERATION: NO                     VOLUME SET(CPU):

   OPEN PARAMETERS: “NWS”
  CLOSE PARAMETERS:
  PRE-OPEN EXECUTE:
POST-CLOSE EXECUTE:
           QUEUING:                   SUPPRESS FORM FEED: YES
_____________________________________________________________________________
Exit     Save     Refresh

Enter a command or ‘^’ followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

16.1.3.2 NULL Devices

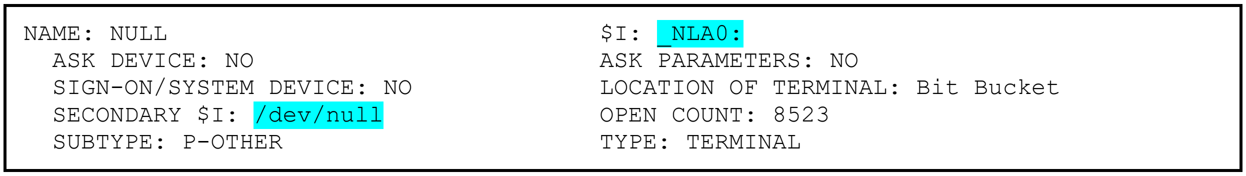
Figure 198 and Figure 199 shows a **NULL** device entry for a mixed operating system, VMS (Primary) and Linux (Secondary), using the **TRM or VTRM Device Edit** [XUDEVEDITTRM] option to update the SECONDARY $I (#52) field in the DEVICE (#3.5) file:

Figure 198: Mixed Operating System: VMS (Primary) and Linux (Secondary) NULL Device—Sample Data Entry Screen



```
                     Edit a TRM or VTRM device

NAME: NULL                                      LOCATION: Bit Bucket

     $I: _NLA0:
 ALT $I: /dev/null
   TYPE: TERMINAL
SUBTYPE: P-OTHER

                                    SIGN-ON/SYSTEM DEVICE: NO
                                          VOLUME SET(CPU):

    ASK DEVICE: NO                                  MARGIN WIDTH:
ASK PARAMETERS: NO                                   PAGE LENGTH:

       QUEUING:                         SUPPRESS FORM FEED:

_______________________________________________________________________________
Exit     Save     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                       Press <PF1>H for help    Insert
```

Figure 199: Mixed Operating System: VMS (Primary) and Linux (Secondary) NULL Device—Sample DEVICE File Entries



```
NAME: NULL                                $I: _NLA0:
  ASK DEVICE: NO                          ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO               LOCATION OF TERMINAL: Bit Bucket
  SECONDARY $I: /dev/null                 OPEN COUNT: 8523
  SUBTYPE: P-OTHER                        TYPE: TERMINAL
```

REF: For additional sample **NULL** device entries, see Section 16.6.4.2, "NULL Device."

16.1.3.3 BROWSER Devices

Figure 200 shows DEVICE (#3.5) file entries for a BROWSER device:

Figure 200: BROWSER Device—Sample DEVICE File Entry

```
NAME: BROWSER                           $I: USER$:[BROWSER]DDBR.TXT
  ASK DEVICE: YES                       ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             QUEUING: NOT ALLOWED
  LOCATION OF TERMINAL: BROWSER         ASK HOST FILE: NO
  ASK HFS I/O OPERATION: NO             SECONDARY $I: /tmp/ddbr.txt
  OPEN COUNT: 1                         OPEN PARAMETERS: (“NWS”)
  POST-CLOSE EXECUTE: D POST^DDBRZIS I $G(IO(“CLOSE”))’=“” N % S %=$$DEL1^%ZISH(
IO(“CLOSE”))
  PRE-OPEN EXECUTE: N X S X=$$TEST^DDBRT S:X IO=$$UNIQUE^%ZISUTL(IO) I ‘X S %ZIS
QUIT=1,X=“Browser not selectable from current terminal.” W $C(7),!,X
  SUBTYPE: P-BROWSER                    TYPE: HOST FILE SERVER
```

16.1.3.4 P-MESSAGE Devices

Figure 201 shows DEVICE (#3.5) file entries for a P-MESSAGE device:

Figure 201: P-MESSAGE Device—Sample DEVICE File Entry

```
NAME: P-MESSAGE-HFS-ONT                 $I: USER$:[TEMP]XMHFS.TMP
  ASK DEVICE: NO                        ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             QUEUING: ALLOWED
  LOCATION OF TERMINAL: HFS FILE==> MESSAGE
  ASK HOST FILE: NO                     ASK HFS I/O OPERATION: NO
  SECONDARY $I: /tmp/xmhfs.tmp          OPEN PARAMETERS: (“NWS”)
  PRE-OPEN EXECUTE: D EN^XMAPHOST Q:$G(POP)  S IO=$$UNIQUE^%ZISUTL(IO)
  SUBTYPE: P-MESSAGE-HFS-ONT            TYPE: HOST FILE SERVER
```

16.1.3.5 TELNET Devices

Figure 202 and Figure 203 show DEVICE (#3.5) file entries for TELNET devices:

Figure 202: TELNET Device—Sample DEVICE File Entry (1 of 2)

```
NAME: TELNET/LINUX                      $I: /dev/pts/
  ASK DEVICE: YES                       SIGN-ON/SYSTEM DEVICE: YES
  LOCATION OF TERMINAL: Telnet Terminal
  OPEN COUNT: 101                       SUBTYPE: C-VT320
  TYPE: VIRTUAL TERMINAL
```

Figure 203: TELNET Device—Sample DEVICE File Entry (2 of 2)

```
NAME: TELNET/VMS                        $I: TNA
  ASK DEVICE: YES                       ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: YES            LOCATION OF TERMINAL: Telnet terminal
  OPEN COUNT: 8657                      SUBTYPE: C-VT320
  TYPE: VIRTUAL TERMINAL
```

16.2 Mixed OS Environment Fields

NOTE: This is for Caché only.

With the advent of remote data centers (RDCs), the VA may use mixed OS environments with a Caché Extended Caché Protocol (ECP) App/Data server configuration. In this environment output devices need different **$IO** values depending on where the job is running. Kernel patch XU*8.0*440 added support to allow the Device Handler to work in a mixed operating system (OS) environment. The fields in Table 29 were added to the KERNEL SYSTEM PARAMETERS (#8989.3) file to provide this support:

Table 29: Mixed OS Environment Fields in the KERNEL SYSTEM PARAMETERS (#8989.3) File

Field	Description
MIXED OS (#.05)	This is used to select which field to use when selecting operating system (OS)-specific data fields in a mixed OS environment. The support is for Caché in an ECP client/server configuration with only two operating systems at a time. In a mixed environment the primary OS is always VMS, and the secondary OS is *not* VMS (i.e., Linux or NT). Some of the fields that need mixed values are: • PRIMARY HFS DIRECTORY (#320) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file • SECONDARY HFS DIRECTORY (#320.2) field in the KERNEL SYSTEM PARAMETERS (#8989.3) file • SECONDARY $I (#52) field in the DEVICE (#3.5) file
SECONDARY HFS DIRECTORY (#320.2)	This field holds the secondary HFS directory path.

Field	Description
LOGICAL DISK NAME (#504)	This field holds a logical disk name that is stored in the Caché CPF file for client system in an ECP client/server configuration.
PHYSICAL DISK (#505)	This field holds the physical disk name to which Cache VMS converts the LOGICAL DISK NAME (#504).
SECONDARY $I (#52)	This field holds the **$IO** value to be used if this is the secondary system in a mixed OS environment. It is *not* used otherwise. It is only used for output devices.

16.2.1 Edit Logical/Physical Mapping Option

The **Edit Logical/Physical Mapping** option [XU SID EDIT] option, which is located on the **Kernel Management Menu** [XUKERNEL] menu, allows you to edit the fields that support the LOGICAL to PHYSICAL translation for the System ID. This is only valid in a Caché 5.2 client/server configuration.

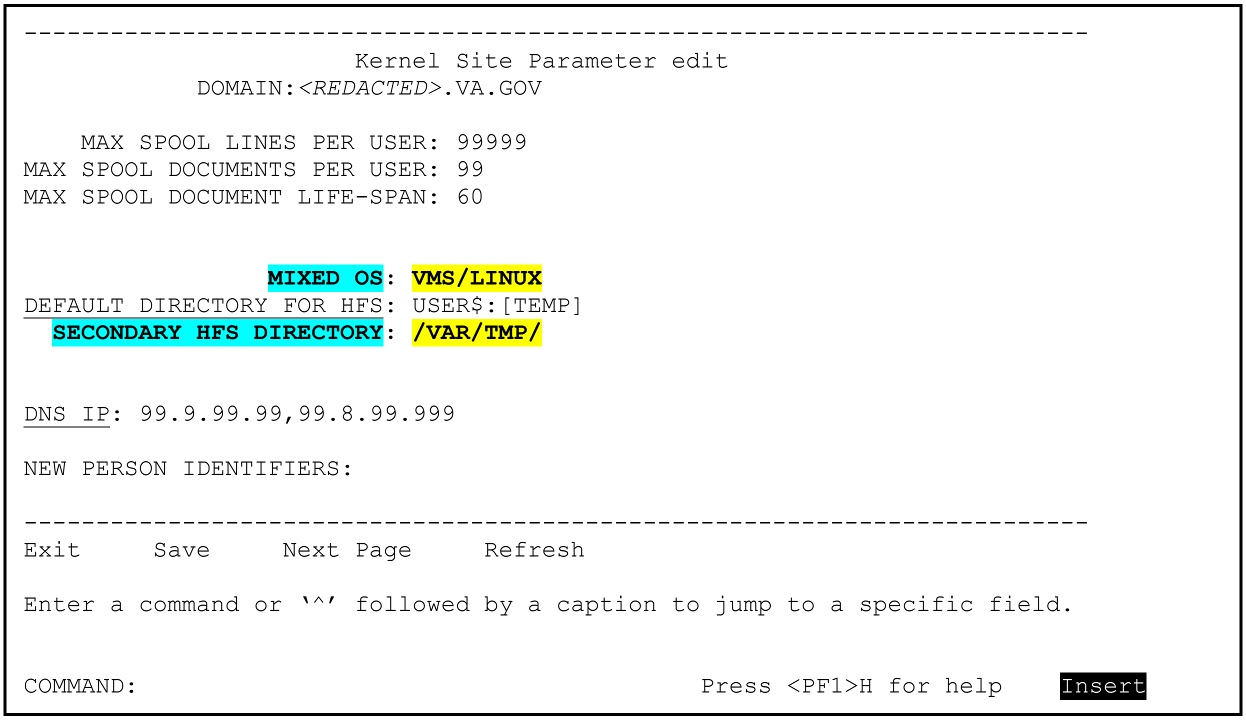
NOTE: This option was released with Kernel Patch XU*8.0*440.

16.2.2 Enter/Edit Kernel Site Parameters option

The **Enter/Edit Kernel Site Parameters** [XUSITEPARM] option's Screen 3 (Figure 204) displays the following fields that were added to the KERNEL SYSTEM PARAMETERS (#8989.3) file:

- MIXED OS (#.05)
- SECONDARY HFS DIRECTORY (#320.2)

Figure 204: Enter/Edit Kernel Site Parameters Option—ScreenMan Form 3: MIXED OS (#.05) and SECONDARY HFS DIRECTORY (#320.2) Fields



```
--------------------------------------------------------------------------
                       Kernel Site Parameter edit
            DOMAIN:<REDACTED>.VA.GOV

    MAX SPOOL LINES PER USER: 99999
MAX SPOOL DOCUMENTS PER USER: 99
MAX SPOOL DOCUMENT LIFE-SPAN: 60


                 MIXED OS: VMS/LINUX
DEFAULT DIRECTORY FOR HFS: USER$:[TEMP]
  SECONDARY HFS DIRECTORY: /VAR/TMP/


DNS IP: 99.9.99.99,99.8.99.999

NEW PERSON IDENTIFIERS:

--------------------------------------------------------------------------
Exit     Save     Next Page     Refresh

Enter a command or '^' followed by a caption to jump to a specific field.


COMMAND:                                              Press <PF1>H for help    Insert
```

NOTE: This option was updated with Kernel Patch XU*8.0*440.

16.3 Device Security

To regulate who can use a particular device, you can use the PASSWORD and SECURITY fields.

The SECURITY field, if populated, should contain a string of characters to compare with a user's FILE MANAGER ACCESS CODE (#3) field, **DUZ(0)**, when the device is selected. Access is denied to anyone whose **DUZ(0)** does *not* contain one of the specified characters. As with other uses of **DUZ(0)**, the at-sign (@; Programmer access) overrides this restriction.

The PASSWORD field, if populated, forces all users trying to log on to the device to be prompted for the matching password, before entering their Access code.

16.4 TERMINAL TYPE (#3.2) File

The TERMINAL TYPE (#3.2) file holds device vendor-specific code to characterize a terminal type. For example, escape sequences can be entered in the OPEN EXECUTE (#6) and CLOSE EXECUTE (#7) fields to set pitch or font. Every device in the DEVICE (#3.5) file *must* be assigned a terminal type, in the SUBTYPE (#3) field.

The most common fields to populate for TERMINAL TYPE (#3.2) file entries are listed in Table 30:

Table 30: Common Fields in the TERMINAL TYPE (#3.2) File

Field	Description
NAME (#.01)	The name of the terminal type. ℹ **REF:** For a description and list of acceptable terminal type name formats, see the "Terminal Type Naming Conventions" section and Table 31.
SELECTABLE AT SIGN-ON (#.02)	This field is used to screen the choices that can be made at the "DEVICE TYPE" prompt during signon.
RIGHT MARGIN (#1)	This field is the number of characters wide for this device.
FORM FEED (#2)	The argument of an M **WRITE** statement that sets the top-of-form for the use of tractor-feed paper on a printer or clears the screen of a video display terminal.
PAGE LENGTH (#3)	This field is the number of usable lines on the output device.
BACK SPACE (#4)	The argument of an M **WRITE** statement that causes the cursor to back space.
OPEN EXECUTE (#6)	This is the executable M code that is used by **%ZIS** to **OPEN** the terminal.
CLOSE EXECUTE (#7)	This is the executable M code that is used by **%ZIS** to **CLOSE** the terminal [i.e., **X ^%ZIS("C")**].

The TERMINAL TYPE (#3.2) file has many more fields where additional specific information for particular terminal types can be entered. Kernel provides the options shown in Figure 205 to facilitate creating and editing terminal types:

Figure 205: Terminal Type Edit Options

```
Device Management ...                                          [XUTIO]
   Terminal Type Edit                                         [XUTERM]
   Change Device’s Terminal Type                            [XUCHANGE]
   List Terminal Types                                        [XULIST]
```

16.4.1 Terminal Type Naming Conventions

The convention for naming terminal types is shown in Table 31:

Table 31: Terminal Type Naming Conventions

Terminal Type	Description
C-	Video terminals (e.g., C-VT320).
PK-	Printers with keyboards.
P-	Printers without keyboards (e.g., P-LASER).
M-	Modems.

The general format is limited to two alphabetic character prefix, followed by a hyphen, and followed by alphanumeric characters.

As mentioned previously (see Section 15.3.1), a spool document name *cannot* use this format; this is so that it can be distinguished from a device subtype in a call to the Device Handler. Confusion could arise since either can be used as the second piece of the device specification. The SPOOL DOCUMENT (#3.51) file has an input transform pattern match that guards against creation of document names in the format of device subtypes.

16.4.2 How Shared Device and Terminal Type Attributes are Used

The DEVICE (#3.5) and TERMINAL TYPE (#3.2) files share attribute fields for RIGHT MARGIN and PAGE LENGTH. If a value is entered for RIGHT MARGIN or PAGE LENGTH in the DEVICE (#3.5) file, it overrides the value from the TERMINAL TYPE [#3.2] file.

When a user selects a device by responding to the device prompt with only the first required piece of information, the device identification, Device Handler retrieves parameters to characterize the device (e.g., RIGHT MARGIN) from the DEVICE (#3.5) file. Furthermore, the Device Handler checks the ASK PARAMETERS (#5) flag for the selected device and, if the flag is set, prompts the user for associated parameters, presenting DEVICE (#3.5) file characteristics as the default. For terminals and virtual terminals (types TRM and VTRM, respectively), the user is prompted for the right margin. For magtape (MT), Sequential Disk Processor (SDP; obsolete), and Host File Server (HFS) devices, they can be prompted for address/parameters with the value of the OPEN PARAMETERS (#19) field (in the DEVICE [#3.5] file) as the default.

REF: For more information on Magtape (MT) devices, see "Special Devices."

For more information on Host File Server (HFS) devices, see Section 17, "Host Files."

16.4.3 Terminal Type Information Retained by User

User can change some terminal type attributes of their signon device by doing either of the following:

- Changing the terminal type during the session with the **Edit User Characteristics** [XUSEREDITSELF] option.
- Selecting a device for direct output.

Kernel uses the **^XUTL** global to hold information about changes made to device characteristics of the home device during a session.

REF: For more information the **^XUTL** global, see the "Menu Manager: System Management" section.

The terminal type established for users at each signon is stored in their NEW PERSON (#200) file entries so that, if necessary, it can be used as a default for the next signon.

16.5 Devices and Signon

16.5.1 Device Selection at Signon and Virtual Terminal Devices

Every interactive user *must* be associated with a device by the Device Handler when they sign onto the VistA system. The device association is done by matching the incoming user's **$I** (#1) field value with the **$I** value of an entry in the DEVICE (#3.5) file.

Historically, it was practical to set up one device entry with a matching **$I** for each physical port. With the move to OpenVMS, however, the **$I** of the user was dynamic, with many thousands of **$I** values possible. The Virtual Terminal device type (VTRM; see Table 26) was created as a way to have one device entry to be used for signon for multiple incoming **$I** values. The Device Handler still checks to see if it can assign a device to an incoming process based on an exact match of **$I** values. If there is no direct match, however, Device Handler checks to see if the *first part* of the user's **$I** value matches the **$I** value of a virtual device entry. This way, a virtual device with a **$I** value of **_TNA** can service all incoming processes whose **$I** values *start* with the string **TNA**.

Virtual devices do *not* need a value in the VOLUME SET(CPU) (#1.9) field; they should have the SIGN-ON/SYSTEM DEVICE (#1.95) field set to **YES**, however, to speed up the signon device selection process.

Common device prefixes on VMS systems that could be used for virtual terminal device entries include:

- **TNA**—Telnet devices
- **RTA**—Remote processes using the **SET HOST** command
- **FTA**—Secure Shell devices

Processes on VMS systems that use Telnet usually have **$I** values beginning with the prefix **TNA**, concatenated with an integer value and a colon (e.g., "**TNA8456:**"). A single virtual terminal device entry whose **$I** value is **TNA** services all such processes.

16.5.2 Terminal Type Selection at Signon

Besides needing a device assigned at signon, users also need a terminal type. As described in the "Signon/Security: System Management" section, Kernel can usually determine the correct subtype without needing to prompt the user by querying the terminal and matching the returned string (if any) with return codes for terminals stored in the DA RETURN CODES (#3.22) file.

If the user is prompted to enter a terminal type, they need to choose one. The list of terminal types from which they can choose is screened by the SELECTABLE AT SIGN-ON (#.02) field in the TERMINAL TYPE (#3.2) file. Users can only choose from entries with this field set to **YES**. This stops users from choosing inappropriate terminal types. The setting of this field does *not* prevent terminal types from being chosen by the DA return code method, however. Make sure that all terminal types appropriate for signon have SELECTABLE AT SIGN-ON (#.02) set to **YES**.

If the Signon/Security system *cannot* supply even a default, the Device Handler selects according to the signon device's subtype.

16.5.2.1 Managing Display Attributes (DA) Return Codes

Figure 206: DA Return Code Edit Option

```
Device Management ...                                               [XUTIO]
   DA Return Code Edit                                          [XU DA EDIT]
```

The DA RETURN CODES (#3.22) file stores entries for the codes returned by different terminals after Kernel prompts for their display attributes at signon. This file then maps Kernel terminal types to the terminal's return codes. This mapping allows sites to set up mappings for new terminals or to map different terminals to a common type. For example, a site could map all codes returned by all DEC VT type terminals to a single C-VT102 type terminal type.

The DA RETURN CODES (#3.22) file is a small static file managed by the **DA Return Code Edit** option [XU DA EDIT]. You can use the **DA Return Code Edit** option [XU DA EDIT] option to automate the population of the DA RETURN CODES (#3.22) file. When you select this option, the terminal you are using is queried and you are shown the terminal's DA code response. You are then prompted for the terminal type and description for this return code. Enter the terminal type name for the terminal you are using. The option updates the DA RETURN

CODES (#3.22) file, and all terminals responding with this code are recognized at signon. You can quickly populate the DA RETURN CODES (#3.22) file by using this option from several different types of terminals.

Kernel pre-populates the DA RETURN CODES (#3.22) file with a set of standard terminal type entries. You may need to add more entries as needed to handle all terminals at your site.

16.6 Troubleshooting

Figure 207: Device Management—Troubleshooting Options

```
SYSTEM MANAGER MENU                                          [EVE]
  Device Management...                                     [XUTIO]
    Loopback Test of Device Port                     [XUTLOOPBACK]
    Send Test Pattern to Terminal                        [XUTTEST]
    Out of Service Set/Clear                               [XUOUT]
```

Kernel provides several options on the **Device Management** [XUTIO] menu to aid with troubleshooting device problems, which are described in the sections that follow.

16.6.1 Loopback Test of Device Port Option

Use the **Loopback Test of Device Port** [XUTLOOPBACK] option to test an **RS-232** serial data line when using a loopback connection on the line. First, disconnect the data line from the device it is attached to (if any). Then, tie pins **2** and **3** of the **RS-232** serial data line together. This is a loopback connection; data sent down pin **2** (transmit) loops back up pin **3** (receive). The **Loopback Test of Device Port** [XUTLOOPBACK] option sends the letters of the alphabet down the data line one at a time, and it attempts to **READ** them back. If both lines are intact, you should see "**ABCDEFGHIJKLMNOPQRSTUVWXYZ**" print on the terminal from which you are testing the data line.

16.6.2 Send Test Pattern to Terminal Option

Use the **Send Test Pattern to Terminal** [XUTTEST] option to send a simple test pattern to a device. This is an easy way to verify whether a device is connected to the system. It lets you choose how many lines of the test pattern to send, and then sends that number of lines to the device. You can confirm on the device end exactly how many lines of the test pattern you receive, which can be useful when troubleshooting printer handshaking problems.

16.6.3 Out of Service Set/Clear Option

You can use the **Out of Service Set/Clear** [XUOUT] option to set a device out of order. It asks you the date on which to put the device out of order. From that date forward, the Device Handler does *not* allow any jobs to use the device (users get a message that the device is out of order). To clear the out of order status, use this option again and delete the out of order date.

16.6.4 Verify HFS and NULL Device Setup *(required)*

16.6.4.1 HFS Device

Verify you have a Host File Server (HFS) device in the DEVICE (#3.5) file named **HFS**. If you have performed KIDS installations on your server before, you probably already have an appropriate **HFS** device set up. If you do *not* have an entry for this device, you *must* create one.

REF: For information on how to create an **HFS** device, see "Host Files."

16.6.4.2 NULL Device

Verify you have a **NULL** device in the DEVICE (#3.5) file named **NULL** (or whose mnemonic is named **NULL**). You can have other devices with similar names, but one device is needed whose name or mnemonic is **NULL**. The subtype should be a "**P-**" subtype (e.g., **P-OTHER**), the margin should be a minimum of **80**, and the page length should be a minimum of **60**. Sample setups:

Figure 208: VMS NULL Device—Sample DEVICE File Entry

```
NAME: NULL                              $I: _NLA0:
  ASK DEVICE: NO                        ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             LOCATION OF TERMINAL: BIT BUCKIT
  SUBTYPE: P-OTHER                      TYPE: TERMINAL
```

Figure 209: Mixed Operating System: VMS (Primary) and Linux (Secondary) NULL Device—Sample DEVICE File Entry

```
NAME: NULL                              $I: _NLA0:
  ASK DEVICE: NO                        ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             LOCATION OF TERMINAL: Bit Bucket
  SECONDARY $I: /dev/null
  SUBTYPE: P-OTHER                      TYPE: TERMINAL
```

Figure 210: Linux NULL Device Example—Caché NULL Device Setup

```
NAME: NULL                              $I: /dev/null
ASK DEVICE: NO                          ASK PARAMETERS: NO
SIGN-ON/SYSTEM DEVICE: NO               LOCATION OF TERMINAL: BIT BUCKIT
SUBTYPE: P-OTHER                        TYPE: TERMINAL
```

Figure 211: Windows NULL Device Example—Caché NULL Device Setup

```
NAME: NULL                              $I: //./nul
  ASK DEVICE: NO                        ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             LOCATION OF TERMINAL: BIT BUCKIT
  SUBTYPE: P-OTHER                      TYPE: TERMINAL
```

Figure 212 is the TERMINAL TYPE (#3.2) file entry that is used by all of the **NULL** device configurations.

Figure 212: NULL Device Example—P-OTHER Terminal Type Setup

```
NAME: P-OTHER                                   RIGHT MARGIN: 132
  FORM FEED: #                                  PAGE LENGTH: 64
  BACK SPACE: $C(8)                             DESCRIPTION: General prntr (132)
```

16.7 Device Identification and Cross-References

Devices can be selected in several ways from the "DEVICE:" prompt. Besides the NAME (#.01) field, three other attributes: **MNEMONIC**, **LOCAL SYNONYM**, and **$I** can also be used to select devices. When **LOCAL SYNONYM** is used, the Device Handler searches the local CPU for a match. Thus, the same **LOCAL SYNONYM** value (e.g., **PRINTER**) can be used to identify several devices, one per CPU.

When editing devices through VA FileMan, two additional fields can be used for lookup:

- VOLUME SET(CPU) (#1.9)
- SIGN-ON/SYSTEM DEVICE (#1.95)

You can separate these values with a period delimiter, as shown in Table 32:

Table 32: Sample Period-delimited Pieces Used for Device Lookup

Period-delimited Piece	Description
CPU	All devices matching CPU.
CPU.$I	All devices matching the CPU and **$I**.
SYS	All SIGN-ON DEVICES.
SYS.CPU	All SIGN-ON DEVICES matching CPU.
SYS.$I	All SIGN-ON DEVICES matching **$I**.
SYS.CPU.$I	All SIGN-ON devices matching CPU and **$I**.

For example, to display all signon devices on CPU "**BBB**", you could do:

Figure 213: Displaying Signon Devices on a Specific CPU—Sample User Dialog

```
Select DEVICE NAME: SYS.BBB
```

To display all signon devices whose **$I** begins with **_TNA** you could do:

Figure 214: Displaying Signon Devices with a Specific $I—Sample User Dialog

```
Select DEVICE NAME: SYS.._TNA
```

The **^%ZIS** global listing in Figure 215 shows the cross-references for a device with a **$I** value of **99** and an internal entry number (IEN) of **251**. It is a SIGN-ON/SYSTEM DEVICE (#1.95) and has a VOLUME SET(CPU) (#1.9) value of **AAA**.

Figure 215: Global Listing for Device Cross-references—$I Value = 99 and IEN = 251

```
^%ZIS(1,“G”,“SYS.AAA.99”,251) = “”
^%ZIS(1,“CPU”,“AAA.99”,251) = “”
^%ZIS(1,“C”,“99”,251) = “”
```

If this device is a virtual terminal with a **$I** of **_TNA** and established as a SIGN-ON/SYSTEM DEVICE (#1.95) but *not* given a VOLUME SET(CPU) (#1.9) value, the cross-reference structure would be as shown in Figure 216:

Figure 216: Global Listing for Virtual Terminal Device Cross-references—$I Value = _TNA and IEN = 251

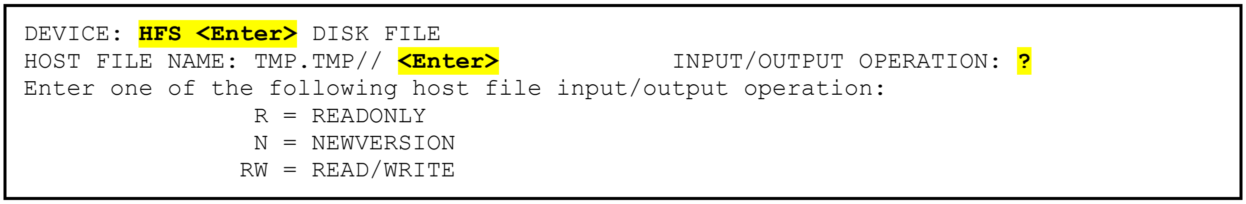
```
^%ZIS(1,“G”,“SYS.._TNA”,251) = “”
^%ZIS(1,“CPU”,“._TNA”,251) = “”
^%ZIS(1,“C”,“_TNA”,251) = “”
```

17 Host Files

17.1 Host Files: User Interface

Host File Server (HFS) devices allow you to send output to a file maintained by your computer's operating system, rather than to a printer. You can send your output to an HFS device, if such a device type has been established on the system. Depending upon how system administrators define the HFS device, you may be prompted for a host file name and for an input/output operation:

Figure 217: Choosing a Host File Server (HFS) Device—Sample User Dialog



```
DEVICE: HFS <Enter> DISK FILE
HOST FILE NAME: TMP.TMP// <Enter>            INPUT/OUTPUT OPERATION: ?
Enter one of the following host file input/output operation:
                R = READONLY
                N = NEWVERSION
               RW = READ/WRITE
```

Not all input/output modes are available on all systems. The possible modes for input/output operation work are shown in Table 33:

Table 33: HFS Input/Output Modes of Operation

Input/Output Mode	Description
APPEND	Data from a **WRITE** operation is appended to the file.
MIXED	Both **READ**s and **WRITE**s are allowed for the specified file.
NEWVERSION	A new file is created with a higher version number; this file can be used for **WRITE**s only.
READ	**READ**s are allowed from the specified file; **WRITE**s are *not* allowed.
READONLY	**READ**s are allowed from the specified file; **WRITE**s are *not* allowed.
READ/WRITE	Both **READ**s and **WRITE**s are allowed for the specified file; if a **WRITE** operation is performed, output is appended to the file.
WRITE	**WRITE**s are allowed; output can be sent to the specified file.

17.2 Host Files: System Management

To provide access to host files through the Device Handler, set up device entries of type HFS.

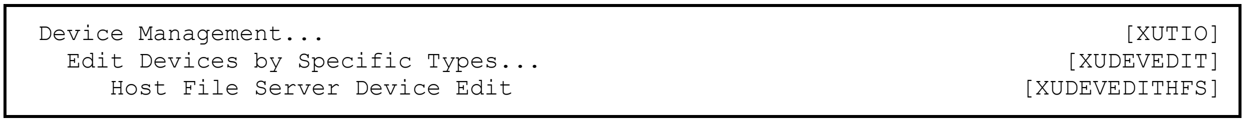
Table 34 lists the three fields in an HFS device entry that act as flags for what a user *must* enter when they use an HFS device:

Table 34: HFS-related Fields in the DEVICE (#3.5) File

Field	Description
ASK PARAMETERS (#5)	If this field is set to **YES**, the user *must* enter the correct M open parameters to open the device. This field should be set to **NO** if the device is accessible to *non*-system administrator users. If it is set to **YES**, the default value is the current value of the OPEN PARAMETERS (#19) field.
ASK HOST FILE (#5.1)	When this field is set to **YES**, the user can choose what file is opened. If it is set to **NO**, the default file name built into the device entry is always used. This field should be set to **NO** if the HFS device is accessible to *non*- system administrator users. Host files can proliferate if too many users are able to create many files. Also, an HFS device opens up access to the host operating system and the potential for overwriting vital files.
ASK HFS I/O OPERATION (#5.2)	If this field is set to **YES**, the user can choose in what mode the file should be opened (e.g., **READ** or **WRITE**). If it is set to **NO**, files are opened in **WRITE** mode. This should be set to **NO** if the device is accessible to *non*- system administrator users, assuming that all such users would only need to **WRITE** host files.

17.2.1 Host File Server Device Edit Option

Figure 218: Host File Server Device Edit Option



```
Device Management...                                            [XUTIO]
  Edit Devices by Specific Types...                         [XUDEVEDIT]
     Host File Server Device Edit                        [XUDEVEDITHFS]
```

The **Host File Server Device Edit** [XUDEVEDITHFS] option lets you to edit Host File Server device attributes using a ScreenMan form.

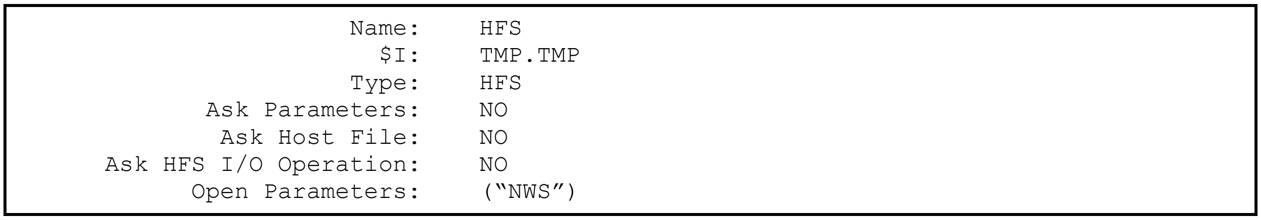
17.2.2 Caché and GT.M HFS Device Setup

Caché and GT.M require the name of the host file to be part of the device **$I** and *not* part of the parameter list.

Table 35: HFS I/O Operation Modes for Caché and GT.M

I/O Operation Mode	Description
NEWVERSION	A new file is created (on VMS with a higher version number); this file can be used for **WRITE**s only.
READONLY	**READ**s are allowed from the specified file; **WRITE**s are *not* allowed.
READ/WRITE	Both **READ**s and **WRITE**s are allowed for the specified file; if a **WRITE** operation is performed, output is appended to the file.

Figure 219: Host File Server Device for Caché and GT.M—Sample Settings



```
                       Name:    HFS
                         $I:    TMP.TMP
                       Type:    HFS
             Ask Parameters:    NO
              Ask Host File:    NO
       Ask HFS I/O Operation:   NO
            Open Parameters:    ("NWS")
```

18 Spooling

18.1 Spooling: User Interface

Spooling privileges can be granted by system administrators to users who prepare and manage reports. By sending your output to the spooler, rather than to a printer, you can benefit in several ways. Since spooling saves the output online in a holding area, you can easily print multiple copies of the report at a later time. Spooling is also a good way to store the results of a time-consuming calculation (e.g., a complex VA FileMan report). By queuing to the spooler, a report involving intensive processing can be done at night or off hours when the system is relatively free. Output can then be printed during the day when the printer can be attended. Finally, when using the spooler, report processing can run to completion without printer problems interfering.

18.1.1 Sending Output to the Spooler

If you have been given the authority to spool, you can send output to the spooler by responding to the “DEVICE:” prompt with the name of the spool device. Devices used for spooling are commonly named SPOOL or SPOOLER.

If you do *not* have spooling privileges and you try to use the spool device, the spooler issues a message that authority has *not* been granted, as shown in Figure 220:

Figure 220: Unable to Send Output to a Spool Device—Sample Message

```
DEVICE: SPOOL

     You aren’t an authorized SPOOLER user.
```

To send output to the spooler with a customized right margin of **96** and page length of **66**, you can use the syntax in Figure 221:

Figure 221: Specifying Spooled Output Margin and Length

```
DEVICE: SPOOL;96;66
```

After requesting the spool device, you are usually prompted for a spool document name, as shown in Figure 222. The prompt is *not* issued, however, if the spool device has been set up to generate the spool document name itself.

Figure 222: Spool Document Name Prompt

```
DEVICE: SPOOL

Select SPOOL DOCUMENT NAME:
```

To skip the “Select SPOOL DOCUMENT NAME:” prompt, you can specify the spool document name at the “DEVICE:” prompt by entering the name in the second semicolon piece. A name entered here is *not* used if the spooler is set up to generate names itself, however. Because of the format used, the Device Handler knows that a spool document name, rather than a device subtype, is being specified. Subtypes begin with one or two letters followed by a hyphen (e.g., **P-DEC**), while spool document names *cannot* (see Section 15.3.1).

Figure 223: Specifying Spool Device and Document Name

```
DEVICE:  SPOOL;MYDOC

DEVICE:  SPOOL;P-OTHER80;MYDOC
```

If the computing environment is composed of several networked processors, you may need to specify where spooling should take place. The spooler on the current CPU should be chosen unless the output is queued.

Figure 224: Spooling Output to a Spool Device on the Same CPU

```
DEVICE:  SPOOL
  1   SPOOL AAA
  2   SPOOL BBB
Choose 1-2>
```

If the output is queued, you can choose a spooler on another CPU and a time to schedule the job to run.

Figure 225: Queuing Output to a Spool Device

```
DEVICE:  Q
DEVICE:  SPOOL BBB
```

Figure 226: Spooler Parameters at the Device Prompt (Summary)

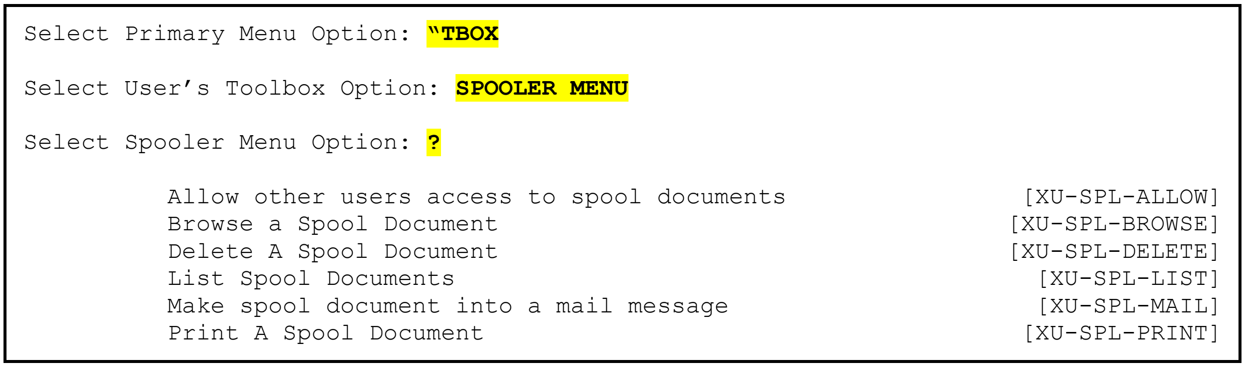
```
DEVICE:  Spooler

DEVICE:  Spooler;Right Margin;Page Length

DEVICE:  Spooler;Subtype

DEVICE:  Spooler;Spool Document Name

DEVICE:  Spooler;Subtype;Spool Document Name
```

18.1.2 Retrieving Spooled Documents

After a spool document has been created, you can retrieve the output by using options on the **Spooler Menu** [XU-SPL-MENU] menu. This menu is distributed as part of Kernel's **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu), a menu available to all users. Specifically, the **Spooler Menu** [XU-SPL-MENU] is in your **User's Toolbox** [XUSERTOOLS] menu.

To quickly reach the Toolbox or any other option on the **Common** menu, you can enter a quotation mark plus the menu text or synonym, as shown in Figure 227:

Figure 227: Spooler Menu Options



```
Select Primary Menu Option: "TBOX

Select User's Toolbox Option: SPOOLER MENU

Select Spooler Menu Option: ?

          Allow other users access to spool documents            [XU-SPL-ALLOW]
          Browse a Spool Document                               [XU-SPL-BROWSE]
          Delete A Spool Document                               [XU-SPL-DELETE]
          List Spool Documents                                    [XU-SPL-LIST]
          Make spool document into a mail message                 [XU-SPL-MAIL]
          Print A Spool Document                                 [XU-SPL-PRINT]
```

18.1.2.1 List Spool Documents Option

The **List Spool Documents** [XU-SPL-LIST] option lists any documents that you have created. Other users *cannot* read or print these documents unless you have authorized them to do so with the **Allow other users access to spool documents** [XU-SPL-ALLOW] option, also on the **Spooler Menu** [XU-SPL-MENU].

18.1.2.2 Delete A Spool Document option

Use the **Delete A Spool Document** [XU-SPL-DELETE] option to delete spool documents. Since there is a limit on the amount of spool space that any one user can consume, you may need to delete old spool documents to free up space for new ones. If you attempt to create a new document when the space limits have been exceeded, the spooler issues a message about the need to delete some documents.

Old documents are deleted automatically, on a schedule as determined by system administrators. System administrators set the "life span" of a spool document via the MAX SPOOL DOCUMENT LIFE-SPAN (#31.3) field in the KERNEL SYSTEM PARAMETERS (#8989.3) File. System administrators should inform you of the life span of spooled documents, so that you are *not* surprised when old documents are purged.

18.1.3 Browsing a Spool Document

18.1.3.1 Browse a Spool Document Option

With the **Browse a Spool Document** [XU-SPL-BROWSE] option, you can view spool documents with VA FileMan’s Browser.

The Browser allows you to:

- View spool documents on your terminal screen.
- Scroll backward and forward through the report.
- Perform simple searches within the report.

REF: For more information on using the Browser, see the *VA FileMan User Manual*.

18.1.4 Printing Spool Documents

18.1.4.1 Print A Spool Document Option

Use the **Print A Spool Document** [XU-SPL-PRINT] option to print spool documents. Before selecting an output device, you are prompted for the number of copies to print. If you have been granted the ability to print to multiple devices, you can send your output to several devices for simultaneous printing. If this privilege has been granted to you, the device prompt is displayed again after you choose the first printer. Entering a **NULL** response to the second device prompt tells the spooler *not* to use any more additional printers.

To save users the time and trouble of despooling their documents, system administrators can set up a spool device for auto-despooling. If you invoke such a spool device, the spool document is sent to one or more printers when the spooling process has completed. After automatic printing, the spool document remains available for reprinting as necessary (it is *not* automatically deleted upon despooling).

18.1.5 Making Spool Documents into Mail Messages

18.1.5.1 Make spool document into a mail message Option

If you have been granted the ability to make spool documents into mail messages, the **Make spool document into a mail message** [XU-SPL-MAIL] option on the **Spooler Menu** [XU-SPL-MENU] is available. You can use it to make documents into regular mail messages that can then be edited, copied, or forwarded just like other VistA MailMan messages. After the text has been moved into a mail message, the spool document is deleted. The deletion is to allow space for new spool documents.

If you plan to make a document into a message, you should do the original output to the spool device with an appropriate margin and page length for a MailMan message. Since MailMan breaks incoming text lines at about the **75**th character, a right margin of **75** may be desirable. Indicating that page breaks should *not* be inserted during the spooling process may also be desirable. Otherwise, the VA FileMan window command **|TOP|** is inserted into the text at the beginning of each page. While this automatic formatting is an advantage when printing spool documents, it is a disadvantage when creating a mail message. Page breaks are *not* inserted when indicating a page length of **99999** lines or a number greater than the document's total. Therefore, when you know your spool document is to be a MailMan message, a suitable margin and page length request might be what is shown in Figure 228:

Figure 228: Formatting/Sending a Document to a Spool Device to Print as a MailMan Message—Sample User Dialog

```
DEVICE: SPOOL;75;99999
```

To turn the spool document into a MailMan message, once your spool document completes, go to the **Spooler Menu** [XU-SPL-MENU] and select the appropriate option, as shown in Figure 229:

Figure 229: Make Spool Document into a Mail Message Option

```
Select Primary Menu Option: ^SPOOLER MENU

Select Spooler Menu Option: MAKE SPOOL DOCUMENT INTO A MAIL MESSAGE
```

If the number of lines in the document exceeds **500**, you are asked whether the transfer process should be queued. This prompt is provided for your convenience since queuing of a time-consuming process is usually preferred. After using the option, you can find your messages by reviewing recently delivered mail in your MailMan IN basket.

18.2 Spooling: System Management

18.2.1 Spool Document Storage

Spool document identification is stored in the SPOOL DOCUMENT (#3.51) file in the **^XMB** global. This file is for internal use by Kernel's spooler and should *not* be directly manipulated by system administrators. It holds identifying information, such as the name of the spool document and the line count totals. The document's text is stored in the SPOOL DATA (#3.519) file in the **^XMBS** global. If the spool document is made into a mail message, the text is moved into the MESSAGE (#3.9) file, the **^XMB** global, and the corresponding entry in the SPOOL DOCUMENT (#3.51) file is deleted.

When initially creating a spooled document, output is sent to the operating system's spooling area (as defined in the spool device). Kernel's spooler moves the output into the **^XMBS** global when the operating system's spooling process is complete. The status of the document (a field in the SPOOL DOCUMENT [#3.51] file) is then changed from Active to Ready and the document can be accessed by the user. Thus, except during spooling, the operating system's spool area should be empty.

18.2.2 Overflowing Spool Document Storage

When the output is moved from the operating system's spool area into the **^XMBS** global, the lines are counted. If during the count the user's maximum line limit is reached (as defined in the MAX SPOOL LINES PER USER [#31.1] field in the KERNEL SYSTEM PARAMETERS [#8989.3] file), the transfer process is halted and a notification message is appended to the transferred text. The entry in the SPOOL DOCUMENT (#3.51) file is also marked as incomplete. Thus, the **^XMBS** global is protected from growth expansion that could overflow the disk storage area.

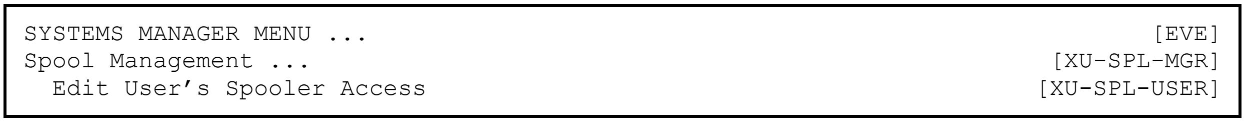
The Kernel spooler *cannot*, however, count the lines of output as they are sent to the operating system's spool area. If the user's line limit is *not* exceeded before initiating the report, Kernel permits sending of an unlimited amount of output to the operating system's spooler. System administrators should consider this when granting spooling privileges. Users who are allowed to spool should be trained accordingly.

Users need to anticipate the results of a process they send to the spooler. If they are *not* sure what to expect, they should be instructed to test the process by sending it directly to an output device. If unexpected results should occur (e.g., an endless loop or meaningless sort), they can interrupt and cancel the process. Users should also be advised about appropriate use of processing time. Methods of efficient VA FileMan searching and sorting should be used when invoking the spooler (just as when printing directly). For example, as described in the VA FileMan documentation, the first sort-by field should be a cross-referenced field when possible and search criteria should be specified with the most likely conditions first.

18.2.3 Granting Spooling Privileges

Options on the **Spool Management** [XU-SPL-MGR] menu can be used to grant spooling privileges to users.

Figure 230: Edit User’s Spooler Access Option



```
SYSTEMS MANAGER MENU ...                                           [EVE]
Spool Management ...                                          [XU-SPL-MGR]
  Edit User’s Spooler Access                                 [XU-SPL-USER]
```

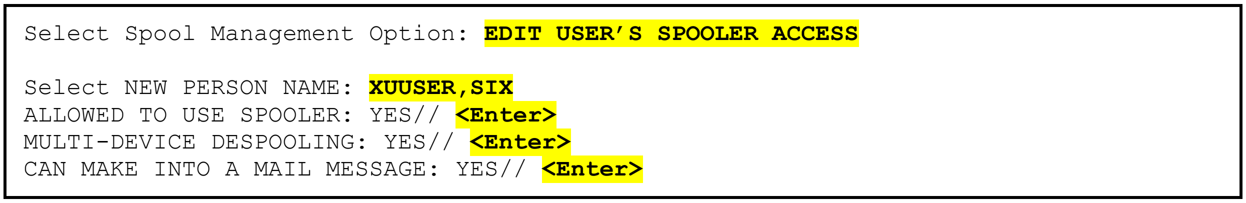
Table 36 lists the spooler-related fields that are user-specific and are stored in the NEW PERSON (#200) file:

Table 36: User Spooler-related Fields in the NEW PERSON (#200) File

Field	Description
ALLOWED TO USE SPOOLER (#41)	If set to **YES** it gives the user the ability to invoke the spooler at the device prompt.
MULTI-DEVICE DESPOOLING (#41.1)	If set to **YES** it enable the user to despool a spooled document to more than one device simultaneously.
CAN MAKE INTO A MAIL MESSAGE (#41.2)	If set to **YES** it permits the conversion of a spool document into a MailMan mail message. The user is able to use all MailMan functions, such as copying and forwarding. As a mail message, the document can no longer be manipulated with the spooler since its flag in the SPOOL DOCUMENT (#3.51) file has been deleted.

As mentioned earlier, the user-oriented spooler options are distributed as part of the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu), a menu available to all users. If system administrators have chosen to lock the **Spooler Menu** [XU-SPL-MENU] or removed it from the **Common** menu, access to the options needs to be re-established for users who are allowed to spool via the **Edit User’s Spooler Access** [XU-SPL-USER] option, as shown in Figure 231:

Figure 231: Edit User’s Spooler Access—Sample User Dialog



```
Select Spool Management Option: EDIT USER’S SPOOLER ACCESS

Select NEW PERSON NAME: XUUSER,SIX
ALLOWED TO USE SPOOLER: YES// <Enter>
MULTI-DEVICE DESPOOLING: YES// <Enter>
CAN MAKE INTO A MAIL MESSAGE: YES// <Enter>
```

18.2.4 Managing Spool Documents

The remaining options on the **Spool Management** [XU-SPL-MGR] menu (Figure 232) are also found on the user-oriented **Spooler Menu** [XU-SPL-MENU]. They are provided on the **Spool Management** [XU-SPL-MGR] menu simply for convenience to system administrators to access any spool document on the system. Users *must* hold the XUMGR security key in order to access all spool documents. Together, these options along with the XUMGR security key permit system administrators to view, print, or delete anyone's spooled documents.

Figure 232: Spool Management Menu Options

```
SYSTEMS MANAGER MENU ...                                            [EVE]
Spool Management ...                                         [XU-SPL-MGR]
  Delete A Spool Document                                 [XU-SPL-DELETE]
  List Spool Documents                                      [XU-SPL-LIST]
  Print A Spool Document                                   [XU-SPL-PRINT]
```

18.2.5 Spooler Site Parameters Edit Option

Figure 233: Spooler Site Parameters Option

```
SYSTEMS MANAGER MENU ...                                            [EVE]
   Spool Management ...                                      [XU-SPL-MGR]
      Spooler Site Parameters Edit                          [XU-SPL-SITE]
```

The **Spool Management** [XU-SPL-MGR] menu also has the **Spooler Site Parameters Edit** [XU-SPL-SITE] option for setting the spooler site parameters (system-wide defaults for the spooler). The initial settings are defined when installing Kernel but can be edited afterwards.

The spooler site parameters control the total number of documents a user can create and the total number of lines for all documents. When the limits are reached, the user *cannot* create new spooled documents.

Table 37 lists the effects of the three spooler site parameter fields:

Table 37: Spooler Site Parameter Fields in the KERNEL SYSTEM PARAMETERS (#8989.3) File

Spooler Site Parameter Field	Description
MAX SPOOL LINES PER USER (#31.1)	This field holds the MAX number of lines of spooled output a user is allowed. If the user has more than this number, then they are *not* permitted to spool any more until some of their spool documents are deleted. This only controls allowing the creation of new spool documents and does *not* terminate a job that is running that has gone over the limit. Recommended value **9999**.
MAX SPOOL DOCUMENTS PER USER (#31.2)	This field limits the number of spool documents that any user can have on the system. Recommended value 1**0-100**.
MAX SPOOL DOCUMENT LIFE-SPAN (#31.3)	This field controls the number of days that a spooled document is allowed to remain in the spooler before deletion by the Purge old spool documents [XU-SPL-PURGE] option that needs to be setup to run in the background.

18.2.6 Purge old Spool documents Option

Figure 234: Purge old Spool documents Option

```
PARENT OF QUEUABLE OPTIONS                                      [ZTMQUEUABLE OPTIONS]
  Purge old spool documents                                            [XU-SPL-PURGE]
```

A spool document is automatically deleted when its life span (in days) is reached. The purge is carried out by the **Purge old spool documents** [XU-SPL-PURGE] option. This option is listed on the **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu along with others that should *not* be invoked interactively but should be scheduled to run through TaskMan.

18.2.7 Defining Spool Device Types

The DEVICE (#3.5) file entries for spooler device types make use of information about the underlying operating system’s spooling mechanism. Examples for several operating systems are provided in the topics that follow.

18.2.7.1 Caché and GT.M

Caché and GT.M use an OpenVMS directory for spooling. As indicated in the VistA Cookbook for VAX sites, the directory should be established with full privileges for System, Owner, Group, and World. The directory specifications are used as the **$I** value.

Figure 235: Spool Device for Caché and GT.M

```
                      Name:   SPOOL
                        $I:   VA1$:[SPOOLER]
                      Type:   SPOOL
                   Subtype:   P-OTHER
```

18.2.8 Spool Device Edit Option

The **Spool Device Edit** [XUDEVEDITSPL] option lets you edit spool device attributes using a ScreenMan form.

Figure 236: Spool Device Edit Option

```
Device Management...                                                        [XUTIO]
  Edit Devices by Specific Types...                                     [XUDEVEDIT]
     Spool Device Edit                                               [XUDEVEDITSPL]
```

NOTE: The type of data entered in the **$I** (#1) and OPEN PARAMETERS (#19) fields depends on the type of M system you are using and the mode of access.

REF: For further details, see your M system manuals.

REF: Examples are provided in the “Defining Spool Device Types” section.

18.2.9 Auto-Despooling

For convenience, spool devices can be defined to ensure that despooling takes place automatically, without user interaction. If the AUTO DESPOOL (#31) field in the DEVICE (#3.5) file is set to **YES**, one copy of the spooled output is sent to each device named in the DESPOOL DEVICES (#32) Multiple field. Having the output automatically despooled saves users the time and trouble of logging on and printing a spool document that may have been created the previous evening. Documents are *not* deleted upon despooling; they remain available to the user for subsequent printing.

Figure 237: Device Edit Option—Sample User Dialog

```
Select Device Handler Option: DEVICE EDIT

Select DEVICE NAME: SPOOL
NAME: SPOOL// ^AUTO D <Enter> ESPOOL
AUTO DESPOOL: 1 <Enter> YES
Select DESPOOL DEVICES:
```

18.2.10 Generating Spool Document Names

Spool devices can be set up to automatically generate the name that identifies the spool document. If the GENERATE SPL DOC NAME (#33) field in the DEVICE (#3.5) file is set to **YES**, users of that device are *not* prompted to enter the spool document name. Also, if the flag is set, any user- or developer-defined name [in **IO("DOC")**] is ignored. The generated name consists of the first **15** characters of the spool device's name, followed by an underscore (_), and followed by the internal entry number (IEN) of the spool document in the SPOOL DOCUMENT (#3.51) file.

Figure 238: Generating Spool Document Name—Sample User Dialog

```
NAME: SPOOL// ^GENERATE SPL DOC NAME
GENERATE SPL DOC NAME: YES
```

19 Special Devices

This section discusses the following special devices and device issues:

- Browser Device
- Form Feeds
- Magtape (MT)
- Network Channel Devices
- Resources
- Sequential Disk Processors (Obsolete)
- Slaved Printers

19.1 Browser Device

19.1.1 User Interface

VA FileMan's Browser allows you to:

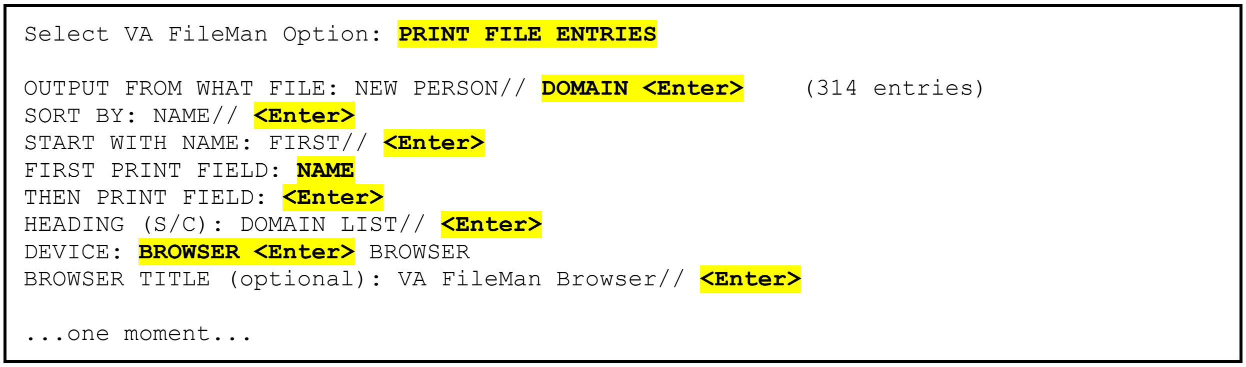
- View reports on your terminal screen.
- Scroll backward and forward through the report.
- Perform simple searches within the report.

If the Browser has been installed at your site and set up as a device, you can use the Browser to view any report that asks you for an output device.

To send a report to the BROWSER device, at any device prompt, enter BROWSER as the device. You may *not* want to send huge reports to the BROWSER, however, since the report *must* complete before you can view its output in the Browser.

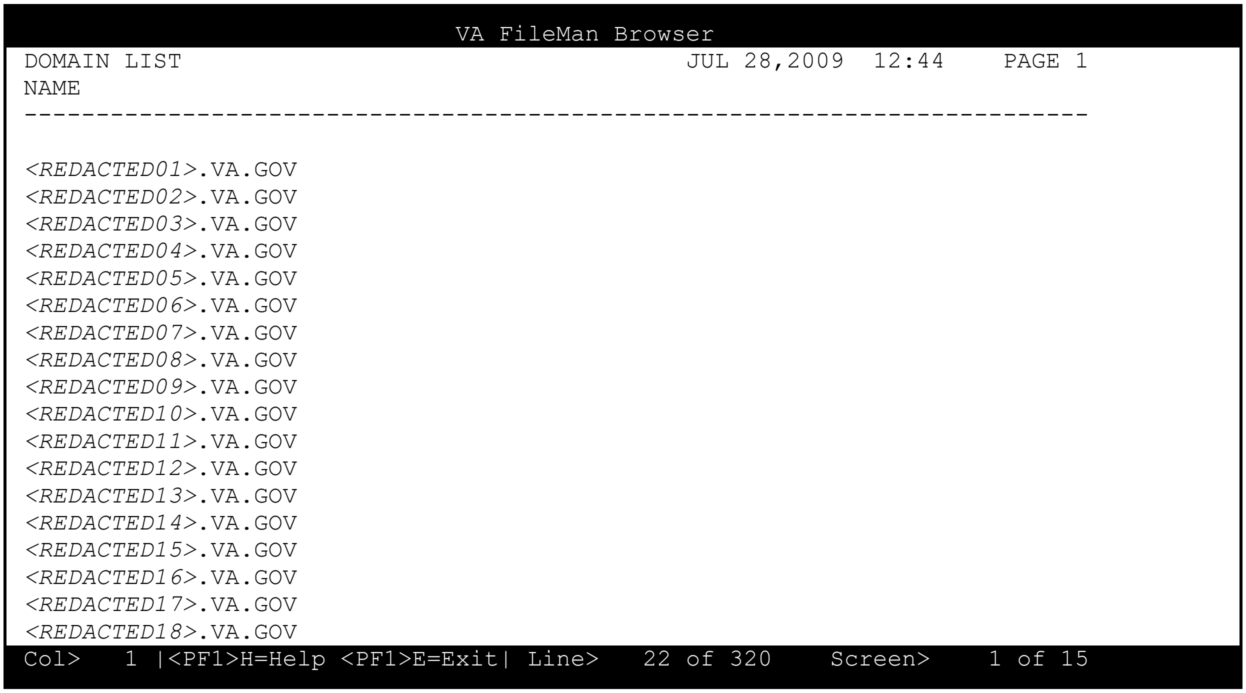
REF: For information on using the Browser and on Browser commands, see the *VA FileMan User Manual*.

Figure 239: Print File Entries Option—Sample User Dialog when Sending a Report to the Browser Device



```
Select VA FileMan Option: PRINT FILE ENTRIES

OUTPUT FROM WHAT FILE: NEW PERSON// DOMAIN <Enter>    (314 entries)
SORT BY: NAME// <Enter>
START WITH NAME: FIRST// <Enter>
FIRST PRINT FIELD: NAME
THEN PRINT FIELD: <Enter>
HEADING (S/C): DOMAIN LIST// <Enter>
DEVICE: BROWSER <Enter> BROWSER
BROWSER TITLE (optional): VA FileMan Browser// <Enter>

...one moment...
```

Figure 240: Print File Entries Option—Sample Domain List report, as Displayed in the Browser Device



```
                              VA FileMan Browser
DOMAIN LIST                                  JUL 28,2009  12:44    PAGE 1
NAME
--------------------------------------------------------------------------


<REDACTED01>.VA.GOV
<REDACTED02>.VA.GOV
<REDACTED03>.VA.GOV
<REDACTED04>.VA.GOV
<REDACTED05>.VA.GOV
<REDACTED06>.VA.GOV
<REDACTED07>.VA.GOV
<REDACTED08>.VA.GOV
<REDACTED09>.VA.GOV
<REDACTED10>.VA.GOV
<REDACTED11>.VA.GOV
<REDACTED12>.VA.GOV
<REDACTED13>.VA.GOV
<REDACTED14>.VA.GOV
<REDACTED15>.VA.GOV
<REDACTED16>.VA.GOV
<REDACTED17>.VA.GOV
<REDACTED18>.VA.GOV
Col>   1 |<PF1>H=Help <PF1>E=Exit| Line>   22 of 320    Screen>    1 of 15
```

19.1.2 System Management

You can set up VA FileMan's Browser as a device to which users can send their output.

When a user sends output to a Browser device, the Browser device performs the following steps:

1. Output is sent to a host file.
2. When the output completes, the host file is closed.
3. The contents of the host file are read back into a scratch global.
4. The host file is deleted.
5. The Browser is called, which displays the data in the global to the user, through the Browser interface.
6. When the user exits the Browser, the scratch global is deleted.

This provides a quick way to generate a report and view the report through the scrollable Browser, potentially saving paper and wear and tear on printers.

To support the Browser device, you need to set up a special terminal type (P-BROWSER) and a special device type (BROWSER).

REF: For sample entries of the special Browser terminal type and device entries for the Caché and GT.M operating systems, see Figure 237 and Figure 238.

The Browser device tests the current terminal to see whether it supports:

- A scrolling region.
- Reverse indexing.

If the terminal does *not* support these features, the Browser device issues a message saying that it is *not* selectable from the current terminal. Also, in order for the check ($$TEST^DDBRT) to work properly, the user *must* already be in the Kernel menu system or *must* have set up developer variables through the ^XUP entry point. Otherwise, the test always fails.

19.1.2.1 Storing Host Files in a Specific Directory

By default, the temporary host files created by the Browser device are stored in the current default directory. You can optionally specify a path to a specific directory to store the temporary host files. Make sure the directory you specify exists on all nodes/CPUs where users can sign on. On DOS systems, do *not* specify the root directory, since there is a limit on the number of files a DOS root directory can hold. Finally, make sure you change both the OPEN PARAMETERS (#19) and POST-CLOSE EXECUTE (#19.8) fields in the Browser DEVICE (#3.5) file entry to specify the directory (replace DD with, for example, D:\BROW\DD).

Figure 241: Caché and GT.M Browser Device—TERMINAL TYPE (#3.2) File Entry

```
NAME: P-BROWSER                         SELECTABLE AT SIGN-ON: NO
  RIGHT MARGIN: 80                      FORM FEED: #
  PAGE LENGTH: 99999                    BACK SPACE: $C(8)
  OPEN EXECUTE: D OPEN^DDBRZIS
  CLOSE EXECUTE: D CLOSE^DDBRZIS
  DESCRIPTION: Browser Device
```

Figure 242: Caché and GT.M Browser Device—DEVICE (#3.5) File Entry

```
NAME: BROWSER                           $I: DDBR.TXT
  ASK DEVICE: YES                       ASK PARAMETERS: NO
  SIGN-ON/SYSTEM DEVICE: NO             QUEUING: NOT ALLOWED
  LOCATION OF TERMINAL: HFS/CRT         ASK HOST FILE: NO
  ASK HFS I/O OPERATION: NO             MARGIN WIDTH: 80
  FORM FEED: #                          PAGE LENGTH: 99999
  BACK SPACE: $C(8)                     OPEN PARAMETERS: NEW:DELETE
  POST-CLOSE EXECUTE: D POST^DDBRZIS
  SUBTYPE: P-BROWSER                    TYPE: HOST FILE SERVER
  PRE-OPEN EXECUTE: I ‘$$TEST^DDBRT S %ZISQUIT=1 W $C(7),!,“Browser not selectable
from current terminal.”,!
```

19.2 Form Feeds

19.2.1 User Interface

Most users would prefer to see their printouts without any extra blank pages before or after the content. Most prefer to see their reports printed on a fresh page instead of starting in the middle of the previous printout. The printing of labels should also be accomplished without unnecessary form feeds. If a printer is generating extra pages, you should contact the system administrators to remedy the problem.

19.2.2 System Management

If a particular device does *not* need a form feed between reports, system administrators should set the SUPPRESS FORM FEED AT CLOSE (#11.2) field to **YES** in the device's DEVICE (#3.5) file entry. Label printers, for example, should have this flag set. This procedure prevents the Device Handler from issuing a form feed, as shown in Figure 243:

Figure 243: Device Edit Option—Sample User Dialog

```
Select Systems Manager Menu Option: DEVICE HANDLER
Select Device Handler Option: DEVICE EDIT

Select DEVICE NAME: LABEL PRINTER
NAME: LABEL PRINTER// ^SUP <Enter> PRESS FORM FEED AT CLOSE
SUPPRESS FORM FEED AT CLOSE: YES
```

The Device Handler also checks the TERMINAL TYPE (#3.2) file to see if form feeds have been suppressed for that terminal type. It checks for the existence of the **IONOFF** variable. Thus, for certain terminal types (e.g., laser printers), system administrators can set this "**no form feed**" variable in the corresponding terminal type's CLOSE EXECUTE (#7) field.

NOTE: The **IONOFF** variable can also be set by the calling application to suppress form feeds.

Figure 244: Terminal Type Edit Option—Sample User Dialog

```
Select Systems Manager Menu Option: DEVICE HANDLER
Select Device Handler Option: TERMINAL TYPE EDIT
Select TERMINAL TYPE NAME: P-DEC-LABEL
NAME: P-ZPK80// ^CLOSE EXECUTE
CLOSE EXECUTE: S IONOFF=""
```

19.3 Magtape

19.3.1 System Management

Figure 245: Edit Devices by Specific Types Option

```
Device Management...                                                  [XUTIO]
    Edit Devices by Specific Types...                             [XUDEVEDIT]
       Magtape Device Edit                                      [XUDEVEDITMT]
```

The **Edit Devices by Specific Types** [XUDEVEDIT] option lets you edit specific types of devices using ScreenMan.

Values entered in a Magtape (MT) device for the fields in Table 38 may *not* be relevant to a given application:

Table 38: Fields in the DEVICE (#3.5) and TERMINAL TYPE (#3.2) Files that May Not be Relevant for Certain Devices

File	Field	Description
DEVICE (#3.5)	SUBTYPE (#3)	Use this field to select a default terminal type for the device. This field points to the TERMINAL TYPE (#3.2) file to retrieve a standard set of characteristics that have been defined for vendor devices (e.g., Laser printers or VT320 CRTs). **REF:** For a discussion of the TERMINAL TYPE (#3.2) file, see the "TERMINAL TYPE (#3.2) File" section.
	MARGIN WIDTH (#9)	Data in this field overrides the RIGHT MARGIN field value from the TERMINAL TYPE (#3.2) file. Leave this field blank unless you are sure that you need to have a different RIGHT MARGIN than what is in the TERMINAL TYPE (#3.2) file.
TERMINAL TYPE (#3.2)	FORM FEED (#2)	The argument of an M **WRITE** statement that sets the top-of-form for the use of tractor-feed paper on a printer or clears the screen of a video display terminal.
	PAGE LENGTH (#3)	This field is the number of usable lines on the output device.
	BACK SPACE (#4)	The argument of an M **WRITE** statement that causes the cursor to back space.

The data values entered in these fields may be arbitrary for Magtape devices. However, if the application plans to copy the output to a printer, the characteristics may need to be similar to that of the printer.

If an application intends to use these fields, be cautious about the type of data that is entered. When sent to the tape unit, some control codes initiate tape movement or cause tape markers to be written to the mounted tape.

Data entered in the **$I** and OPEN PARAMETERS fields depends on the following:

- Type of M system you are running.
- Type of tape unit.
- Desired format.

REF: For examples of the type of data required in these fields, see the "Device Handler: System Management" section.

REF: For further details on Magtape devices, see your specific M implementation manuals.

19.4 Network Channel Devices

19.4.1 System Management

Network channel devices are typically high-speed channel devices (e.g., TCP/IP). Currently, this network channel device support exists under the Caché and GT.M operating system. In most cases, these devices are used for specialized purposes rather than for general output. For example, network mail could use such devices to move enormous amounts of email through high speed communication channels.

The use of network channel devices requires at least two processes on each end of the communication channel, a server and a client, which can then exchange information:

- Server Process—One process *must* be available at all times. It can be actively running or triggered to run at a given moment. This process is commonly known as a server. The server waits until another process makes a request to exchange information.
- Client Process—The other process is known as the client.

The two processes can be hosted by two CPUs using network protocols.

19.4.1.1 Network Channel Device Edit

Figure 246: Network Channel Device Edit Option

```
Device Management...                                         [XUTIO]
    Edit Devices by Specific Types...                    [XUDEVEDIT]
       Network Channel Device Edit                   [XUDEVEDITCHAN]
```

The **Network Channel Device Edit** [XUDEVEDITCHAN] option allows you to edit network channel device attributes.

When editing Network Channel devices, the contents of the fields listed in Table 38 are *not* necessarily relevant for using network Channel devices. However, these fields are provided in case the application calling the Device Handler is *not* able to distinguish between a printer and a Network Channel device when sending output.

The timeout on the M **OPEN** command may *not* be applicable with Network Channel devices. Therefore, it may be necessary to answer **NO** to the USE TIMEOUT ON OPENS (#2009.5) field in the DEVICE (#3.5) file.

REF: For more information regarding device timeout applicability, see the appropriate Caché manual.

For Network Channel devices that use TCP/IP, data is required for the OPEN PARAMETERS (#19) field in the DEVICE (#3.5) file. For the client device setup, this field stores the remote Internet address to which the host connects.

Figure 247: Network Channel Device Edit Option—Sample Output

```
                             EDIT A NETWORK CHANNEL DEVICE
NAME: SDD-DIRECT                                                    PAGE 1 OF 1
_______________________________________________________________________________

NAME: SDD-DIRECT                    LOCATION OF TERMINAL: HP-8000 near Raul
  $I: |TCP|9100                          VOLUME SET(CPU):
TYPE: NETWORK CHANNEL              SIGN-ON/SYSTEM DEVICE: NO

SUBTYPE: P-HP8000 TCP/S                    MARGIN WIDTH:
                                            PAGE LENGTH:

           ASK DEVICE: NO           USE TIMEOUT ON OPENS:
       ASK PARAMETERS: NO                   OPEN TIMEOUT:
      OPEN PARAMETERS: ("10.6.21.138":9100:"M")

             USE LOCK:
```

The GLOBAL LOCK (#36) field in the DEVICE (#3.5) file stores a **YES/NO** Set of Codes. This is important, especially if the application expects that only one client at a given time is able to open the device. If this field is set to **YES** an M lock on **^%ZIS("lock",IO)** is obtained before the device is opened. It remains until a call to ^%ZISC to close the device. It can be used with any type of device.

19.5 Resources

19.5.1 System Management

A Resource device is a type of device that can only be used by tasks. They *cannot* be used for input or output (**I/O**). As such, they are *not* available for user selection at the device prompt. The purpose of a resource is to provide a mechanism of limiting the number of concurrent jobs that can run at any one time.

When creating a task, a task can request the resource as an input variable for the ^%ZTLOAD call. The resource itself, as defined in the DEVICE (#3.5) file, has a field called RESOURCE SLOTS (#35) that determines how many jobs can simultaneously own it as a resource.

The Device Handler and TaskMan work together to provide resource device functionality. The RESOURCE (#3.54) file, stored in the translated **^%ZISL** global, regulates processing and is for internal use only. The NAME (#.01) field holds the **$I** of the resource device. Other fields hold information on jobs currently using the resource, information that is cleared when the resource is closed.

The RESOURCE (#3.54) file supports processing by maintaining a count of the number of available “slots.” The ability to open and close resources is accomplished by decrementing and incrementing this count.

19.5.1.1 Limiting Simultaneous Running of a Particular Task

Resources make it possible for you to control the number of a particular kind of non-I/O task that runs at any one time. If you have a particular job and you want no more than three running versions of it at any one time, you can queue the job (through the ^%ZTLOAD interface) to a resource that had a RESOURCE SLOTS (#35) setting of **3**.

19.5.1.2 Running Sequences of Tasks

Resources also make it possible to run non-I/O tasks in sequential order. Non-I/O tasks ordinarily can run simultaneously because they do *not* compete for the ownership of I/O devices. If you instead queue such tasks to the same resource, and the resource has a RESOURCE SLOTS (#35) setting of 1, TaskMan runs the tasks one at a time and in the order queued. In this way, the results of one process can be used by another. This sequential processing might be appropriate, for example, for the processing of physician orders or other nested tasks involving code execution.

An additional enhancement to resource devices are SYNC FLAGs, which are stored in the SYNC FLAG (#87) field in the TASKS (#14.4) file, allows TaskMan to run the next task waiting for a resource only if the previous task using that resource has completed successfully. You can use SYNC FLAGs to ensure that subsequent jobs run only if previous jobs have completed successfully.

19.5.1.3 Creating Resource Devices

Figure 248: Resource Device Edit Option

```
SYSTEMS MANAGER MENU ...                                             [EVE]
   Device Management ...                                           [XUTIO]
      Resource Device Edit                                  [XUDEVEDITRES]
```

The **Resource Device Edit** [XUDEVEDITRES] option provides a facility for editing resource devices. Software that uses a resource should include in its installation instructions the way the new resource should be defined in the DEVICE (#3.5) file. System administrators can then create one or more resource-type (RES) entries.

Figure 249: Resource Device—Sample Output

```
NAME:  ZZRES       $I:  ZZRES
   LOCATION OF TERMINAL:  NA   RESOURCE SLOTS:  1
   TYPE:  RESOURCE
```

The installation instructions should indicate the number of resource slots. Sequential processing should use a value of 1. The NAME and **$I** should probably use the same value and be namespaced according to VistA conventions.

19.6 Sequential Disk Processors (Obsolete)

Though the Sequential Disk Processors (SDP) entry is still found in the DEVICE (#3.5) file, it is obsolete and users should now use Host File Server (HFS) devices.

REF: For more information on HFS devices, see "Host Files."

19.7 Slaved Printers

19.7.1 User Interface

If your terminal has an auxiliary printer port with a printer directly attached, you can send output normally destined for the CRT terminal directly to a printer. Output for the terminal is redirected from the host computer through the terminal’s auxiliary port to the printer. Such printers are commonly called slaved printers or slaved devices.

If slaved printing is available from your terminal, you can send a printed report to your slaved printer, by entering the device name that corresponds to your slaved printer, as shown in Figure 250:

Figure 250: Slaved Printer—Sample User Dialog

```
DEVICE: SLAVELA50
```

NOTE: Consult your local system administrators to find out if slaved printing devices are available.

19.7.2 System Management

There are two modes of slaved printing:

- Auto Print Mode (a.k.a. Copy Print Mode)—When Auto Print Mode is toggled on, output is displayed on the terminal as well as printed on the printer. Special escape sequences and control characters, such as those that are normally used to adjust fonts/pitches, are *not* passed to the printer; however, those used for actions like carriage return, line feed, and form feed are passed on to the printer.
- Printer Controller Mode (a.k.a. Transparent Print Mode**)**—When Printer Controller Mode is toggled on, output is only printed on the printer; nothing is displayed on the terminal. All escape sequences and control characters are passed to the printer. This mode is preferable to Auto Print Mode, especially when compressed mode printing is desired.

Table 39 lists the escape sequences used to toggle the slaved printing modes for DEC VT220/VT320 terminals:

Table 39: Escape Sequences Used to Toggle the Slaved Printing Modes for DEC VT220/VT320 Terminals

Mode	Escape Sequence
Auto print mode on.	ESC [?5i
Auto print mode off.	ESC [?4i

Mode	Escape Sequence
Printer controller mode on.	ESC [5i
Printer controller mode off.	ESC [4i

19.7.2.1 Device and Terminal Type File Entries

To use a slaved printer through the Device Handler, two DEVICE (#3.5) file entries along with corresponding TERMINAL TYPE (#3.2) file entries *must* be made for the following:

- Home Device
- Slaved Printer

One pair of DEVICE/TERMINAL TYPE entries is needed to describe the home (i.e., CRT) terminal attributes including the codes to open and close the printer port. The OPEN PRINTER PORT (#110) and CLOSE PRINTER PORT (#111) fields of the TERMINAL TYPE (#3.2) file can be used to store the appropriate codes.

Another pair of DEVICE/TERMINAL TYPE entries is needed to describe the attributes of the slaved printer including escape codes to adjust fonts/pitches. The OPEN EXECUTE (#6) and CLOSE EXECUTE (#7) fields of the TERMINAL TYPE (#3.2) file can be used to hold such codes. Additionally, the device entry for the slaved printer *must* have a value of **0** (**zero**) entered into the **$I** field. This **$I** value identifies the DEVICE (#3.5) file entry as one for a slaved device.

The examples in Figure 251 through Figure 256 show the setup for a home device, and the setup for slaved printers.

Figure 251: Home Device Example (VT320)—DEVICE (#3.5) File Entry

```
NAME: TELNET DEVICE                         $I: _TNA
  ASK DEVICE: YES                         ASK PARAMETERS: NO
  VOLUME SET(CPU): KDE                    SIGN-ON/SYSTEM DEVICE: YES
  LOCATION OF TERMINAL: Network           MARGIN WIDTH: 80
  FORM FEED: #,$C(27,91,50,74,27,91,72)  PAGE LENGTH: 24
  BACK SPACE: $C(8)                       SUBTYPE: C-VT320
  TYPE: VIRTUAL TERMINAL
  TIMED READ (# OF SECONDS): 400
```

Figure 252: Home Device Example (VT320)—TERMINAL TYPE (#3.2) File Entry

```
NAME: C-VT320                           SELECTABLE AT SIGN-ON: YES
  FORM FEED: #,$C(27,91,50,74,27,91,72)  RIGHT MARGIN: 80
  PAGE LENGTH: 24                         BACK SPACE: $C(8)
  DESCRIPTION: Digital Equipment Corporation VT-320 video
  OPEN PRINTER PORT: W *27,” [5i”
  CLOSE PRINTER PORT: W *27,” [4i”
```

Figure 253: Slaved Printer Example: DEC LA50—DEVICE (#3.5) File Entry

```
NAME: SLAVELA50                        $I: 0
  ASK DEVICE: YES                      ASK PARAMETERS: YES
  SLAVED FROM DEVICE: TRM
  LOCATION OF TERMINAL: SLAVE DEVICE FOR LA50
  MARGIN WIDTH: 132                    FORM FEED: #
  PAGE LENGTH: 64                      SUBTYPE: P-LA50
  TYPE: TERMINAL
```

Figure 254: Slaved Printer Example: DEC LA50—TERMINAL TYPE (#3.2) File Entry

```
NAME: P-LA50                           RIGHT MARGIN: 132
  FORM FEED: #                         PAGE LENGTH: 64
  OPEN EXECUTE: W *27,”[4w”            CLOSE EXECUTE: W *27,”[0w”
  DESCRIPTION: LA50 132 COL/16.5 CPI
```

Figure 255: Slaved Printer Example: Epson LQ870—DEVICE (#3.5) File Entry

```
NAME: SLAVELQ870                       $I: 0
  ASK DEVICE: YES                      ASK PARAMETERS: YES
  SLAVED FROM DEVICE: TRM
  LOCATION OF TERMINAL: SLAVE DEVICE FOR LQ870
  MARGIN WIDTH: 132                    FORM FEED: #
  PAGE LENGTH: 64                      SUBTYPE: P-LQ870
  TYPE: TERMINAL
```

Figure 256: Slaved Printer Example: Epson LQ870—TERMINAL TYPE (#3.2) File Entry

```
NAME: P-LQ870                          RIGHT MARGIN: 132
  FORM FEED: #                         PAGE LENGTH: 64
  OPEN EXECUTE: W *15                  CLOSE EXECUTE: W *18
  DESCRIPTION: EPSON LQ870 PRINTER--CONDENSED
```

19.7.2.2 Use of Slaved Printer: Processing Steps

The Device Handler manages output to slaved printers using the following steps:

1. Execute the OPEN PRINTER PORT (#110) code of the home device’s terminal type.
2. Execute the OPEN EXECUTE (#6) code of the slaved printer’s terminal type.
3. When the application closes the device, execute the CLOSE EXECUTE (#7) code of the slaved printer’s terminal type.
4. Execute the CLOSE PRINTER PORT (#111) code of the home device’s terminal type.

19.7.2.3 Queuing to Slaved Printers

If queuing to a slaved device is desired, then the SLAVE FROM DEVICE field of the DEVICE (#3.5) file *must* be used. This field is a pointer to the DEVICE (#3.5) file. Data *must* be entered in this field for the entry for the slaved printer. This data should point to the home device entry unless the slaved printer is attached to a terminal on a Terminal Server (i.e., a virtual terminal).

If queuing to a slaved device is being performed from a virtual terminal, then a third device entry *must* be established that fully describes the home device with a type of TRM. This device should be entered into the SLAVE FROM DEVICE field.

NOTE: When queuing to a slaved device from a terminal on a Terminal Server, the user *must* be fully logged off the computer system and logged off the port by the time the queued task is scheduled to run.

IV. TaskMan

20 TaskMan: User Interface

The Kernel TaskMan (TM) software allows you to run tasks (e.g., VA FileMan prints and sorts) in the background and lets you continue working without interruption.

20.1 Creating Tasks

VistA runs in a multiprocessing environment, which means the computer can work on more than one job at a time. Each job the computer works on consumes a part of the computer’s resources. Initially, you have only one job, your interactive terminal session, with which to do your work. TaskMan, however, allows you to claim more of the computer’s resources by allowing you to schedule additional jobs to run in the background.

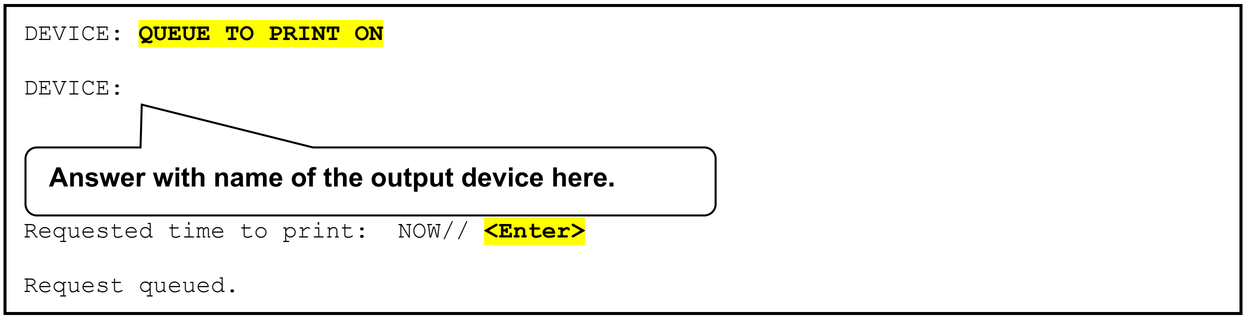
20.1.1 Background Jobs

You can queue additional tasks to run through TaskMan. Once started, these additional tasks (called background tasks) can run at the same time as the foreground jobs and without further dialog with the people who started them. Appropriate use of background tasks can cut your frustration by reducing the amount of time you *must* wait for the computer to do lengthy, repetitious work that does *not* need human intervention. Every task queued to run in the background reduces time spent waiting and also uses the computer’s resources more efficiently.

20.1.2 Queuing Output

Most users use TaskMan by queuing reports, labels, and other kinds of output. Because output involves no dialog once it has begun and because it requires you to wait while it prints, it makes an ideal candidate for queuing. You can queue most output when the computer asks you to select a device to which the output should be sent. The series of prompts and responses to queue a job to a device usually looks something like Figure 257:

Figure 257: Queuing Output—Sample User Dialog



```
DEVICE: QUEUE TO PRINT ON

DEVICE:
```

Answer with name of the output device here.

```
Requested time to print:  NOW// <Enter>

Request queued.
```

After you answer this series of prompts, the output is queued for TaskMan to start at the requested time, and you can continue with other work while TaskMan prints the output. When many tasks need the same device at the same time, TaskMan runs them in order based on the time they were requested.

20.1.3 Other Sources of Tasks

An application can create other kinds of tasks without your interaction. The application might offer to queue other kinds of work like large filing or complex data analysis jobs. Sometimes applications queue tasks without asking. For example, the delivery of MailMan messages is performed by a job running as a task. If that task is *not* running when someone uses the MailMan options, MailMan automatically uses their foreground job to queue the task without asking them. Although people may knowingly or unknowingly queue these other kinds of tasks, output remains the most common kind of work to queue.

20.2 Working with Tasks

Figure 258: TaskMan User Option

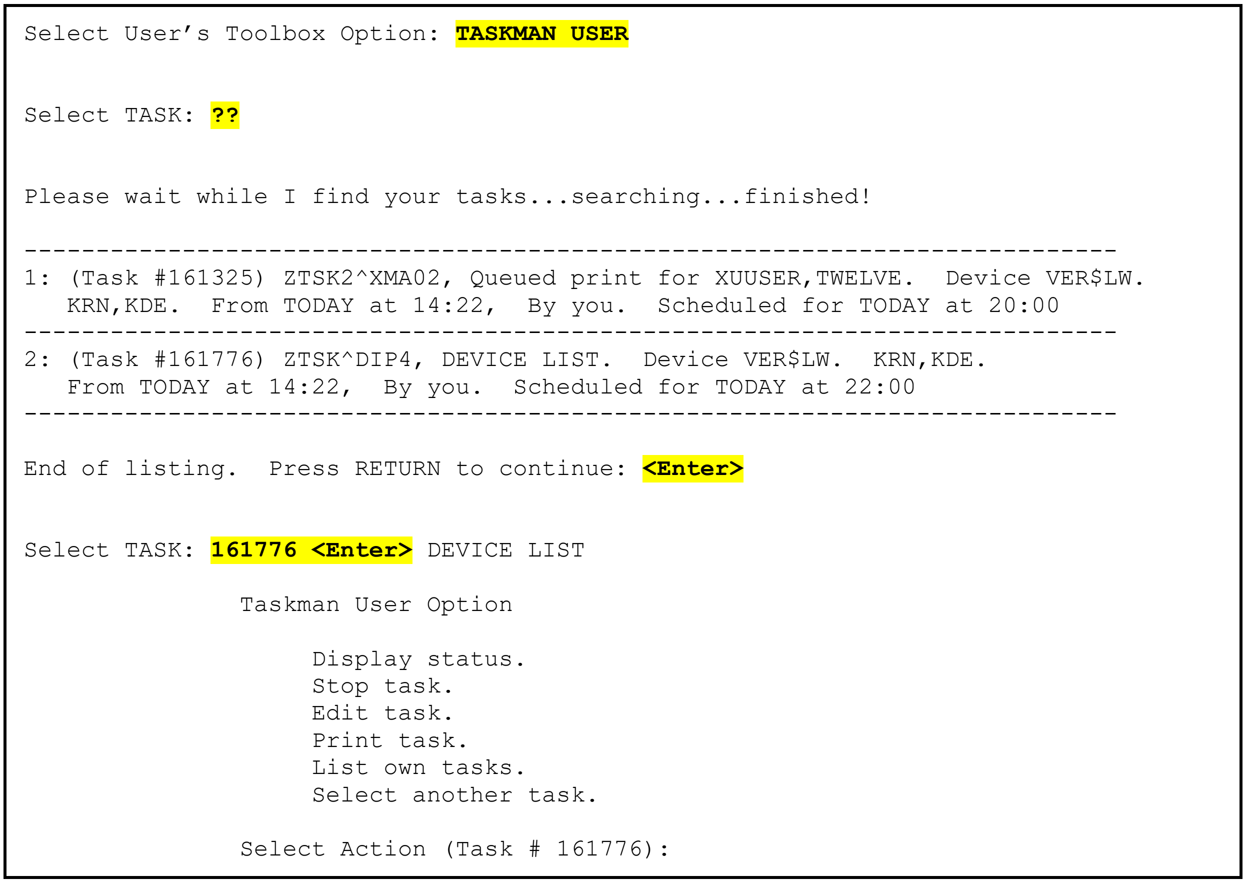
```
System Command Options ...                                        [XUCOMMAND]
User’s Toolbox ...  “TBOX”                                      [XUSERTOOLS]
   TaskMan User                                                  [XUTM USER]
```

TaskMan also allows you to examine or modify your own tasks. You can do this by using the **TaskMan User** [XUTM USER] option, located in the **User’s Toolbox** [XUSERTOOLS] menu on your **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). This option lets you monitor or manipulate one task at a time.

20.2.1 Selecting Tasks

When you choose the **TaskMan User** [XUTM USER] option, it first asks you to select a task with which to work. TaskMan displays the “Select TASK:” prompt. If you enter a single question mark (**?**), you get some general help about the option; if you enter two question marks (**??**), you can get a list of every task that you have queued to run. Typically, you would enter two question marks at this prompt so that you can get a listing of your individual tasks, listed by task number. You then choose a task from the list of tasks to work with. Using the **TaskMan User** [XUTM USER] option is shown in Figure 259:

Figure 259: TaskMan User Option—Sample User Dialog



```
Select User’s Toolbox Option: TASKMAN USER


Select TASK: ??


Please wait while I find your tasks...searching...finished!

----------------------------------------------------------------------------
1: (Task #161325) ZTSK2^XMA02, Queued print for XUUSER,TWELVE.  Device VER$LW.
   KRN,KDE.  From TODAY at 14:22,  By you.  Scheduled for TODAY at 20:00
----------------------------------------------------------------------------
2: (Task #161776) ZTSK^DIP4, DEVICE LIST.  Device VER$LW.  KRN,KDE.
   From TODAY at 14:22,  By you.  Scheduled for TODAY at 22:00
----------------------------------------------------------------------------

End of listing.  Press RETURN to continue: <Enter>


Select TASK: 161776 <Enter> DEVICE LIST

               Taskman User Option

                    Display status.
                    Stop task.
                    Edit task.
                    Print task.
                    List own tasks.
                    Select another task.

               Select Action (Task # 161776):
```

You can select tasks either by task number or list number. In the list of tasks, the list number is at the left-hand side of each task listing and is followed by the task number for each task (in parentheses). The rest of the information helps identify where the task came from and what it does.

20.2.2 Tasks in the Task List

You can only select tasks that are still in TaskMan’s task list. When a task finishes running, it usually removes itself from the task list. Thus, you should *not* get a listing of every task you have run in the last year! Tasks that do *not* clean up their entries usually get cleaned out by TaskMan several days after they complete. You should only have to select tasks that are still actively waiting to start, currently running, or encountered some kind of problem while running.

20.2.3 Display Status of Tasks

Once you have selected a task to work with, you can ask to see the status of that task, using the **Display status** (**D**) action. TaskMan uses a task's status to try to explain how soon the task runs and why. The possible normal statuses for a task include:

- Scheduled for <*date and time*>.
- Being inspected by TaskMan.
- Waiting for a partition.
- Being prepared.
- Currently running.
- Completed <*date and time*>.

NOTE: Please keep in mind that TaskMan can only "guess" whether a task is currently running.

One of the following messages may show up if the task needs some system resource *not* currently available:

- Waiting for device <*name of device*>.
- Waiting for the link to <*name of CPU*> to be restored.

When you display the status of a task waiting for a device, TaskMan shows you how many tasks are in line for that device ahead of your task. Additional statuses exist for tasks that have encountered some kind of problem. For each situation it lists a different explanation of the problem. For example, if you use the Stop task option to stop a task, its status shows up as "Stopped by you."

20.2.4 Stopping Tasks

Under certain conditions, you may want to stop a task. The **TaskMan User** [XUTM USER] allows you to do this through the **Stop task** (**ST**) action. Your ability to stop a task depends on the task's status, however. If the task has already been stopped, is finished, or it encountered a problem while running and you try to stop it, the Stop task option tells you that the task has already stopped. If the task has *not* yet started running, on the other hand, you can always stop it. If the task has started running, the **Stop task** action succeeds in stopping it only if the developer who wrote the task has designed the task to be stopped by a user. At any rate, it does *not* cause any problems if you try to stop a running task.

To stop a task, use the **Stop task** action. Once you stop a task, it remains in the TASKS (#14.4) file until you edit it to run again or until TaskMan purges it from the Task list.

20.2.5 Editing Tasks

The **Edit task** (**E**) action lets you edit a task's output device, description, and run time.

The task *must* be unscheduled before it can be edited. The **Edit task** action asks if it's OK to unschedule the task. To edit the task, answer **YES**. But once the task is unscheduled, it does *not* run unless you reschedule it by finishing each step of editing the task.

NOTE: You *cannot* edit a task that is already running.

Once the task is unscheduled, you can update the following task settings:

- When the task should start.
- Which device it should use (and whether a device is needed).
- What the description of the task should be.

Once you have had a chance to modify these three settings, you are asked whether the task should be rescheduled as shown (see Figure 260):

- If you answer **YES**, the task is updated to reflect the changes you specified.
- If you answer **NO**, however, no settings are changed, but the task remains unscheduled (and does *not* run until you use Edit Task to reschedule it).

Figure 260: Edit Task Option—Sample User Dialog

```
Before you edit the task I'll make sure it's not scheduled, okay?  YES// <Enter>
  Task ready for editing.

  Currently, this task requests output device VER$LW.
  Do you want to change the output device for this task? NO// Y
  Select Task's Output Device (^ for none): P236

  When should this task run?:  AUG 16, 2004@22:00// <Enter>

  Task's purpose: DEVICE LIST// <Enter>

161776: DEVICE LIST.  P236.  Next run time: AUG 16, 2004@22:00.

  Shall I reschedule this task as shown? YES// <Enter>
  Task rescheduled.
```

20.2.6 Listing and Printing Tasks

You can use the **List own tasks** (**L**) action to review your tasks. This option displays the same list as that given when you enter two question marks (**??**) at the "Select Task:" prompt.

The **Print task** action allows you to print out the description of the task that you have currently selected.

20.2.7 Selecting Another Task

Once in the **TaskMan User** [XUTM USER] option, you can choose to work with a different task by using the **Select another task** (**SE**) action. Enter another task number to work with a different task. If you are *not* sure what task you want to work with, you can get a list of all of your tasks by entering two question marks (**??**).

20.3 Summary

Most output in VistA is performed by creating tasks that run in the background. Once you become familiar with TaskMan's queuing system, you can increase productivity by using some of TaskMan's special features, including:

- Listing your future tasks.
- Displaying a task's status.
- Stopping a running task.
- Editing a future task's run time and output device.

21 TaskMan: System Management—Overview

Kernel's TaskMan module provides a standardized system for initiating and managing background processing. Since TaskMan handles all background processes, system managers have a unified set of controls that apply to all background processes on their systems.

Most of TaskMan's processing does *not* involve interaction with users, rendering its operation virtually invisible. The explanations that follow provide information about the operation of TaskMan.

21.1 TaskMan's Division of Labor

TaskMan uses a three-step system to start and manage background processing:

1. **Queuers**

   Foreground jobs *cannot* directly start any background jobs. Instead, they call the TaskMan Application Program Interface (API) to file requests in the TASKS (#14.4) and SCHEDULE files. The program code calling the TaskMan API is called a Queuer. The TASKS (#14.4) file is VA FileMan-compatible. The SCHEDULE file is *not* VA FileMan compatible.

   REF: For a description of the TASKS (#14.4) and SCHEDULE file structure, see the "Troubleshooting" section in the "TaskMan: System Management—Operation" Section.

2. **Manager**

   A TaskMan program called the **manager** runs at all times in the background. The **manager** monitors the SCHEDULE file. As needed, it initiates background jobs (called **submanagers**) to perform the work requested by the foreground jobs.

3. **Submanagers**

   Each background job request is picked up by a TaskMan process called the **submanager**. The **submanager** is the job that actually runs each task. Submanagers handle contention for partitions and **I/O** devices by running the waiting tasks in order, first the oldest tasks and then the more recent ones.

21.1.1 Queuers

Tasks run by TaskMan begin with code in a software application that decides to perform some work in the background. This code is a queuer. Most applications in VistA respond to a user's request to queue some output, but other decisions may be involved. Two commonly used queuers are programs that create report output (by using the TaskMan API) and options that are scheduled through the OPTION SCHEDULING (#19.2) file.

21.1.1.1 Programs that Use the TaskMan API

One commonly used queuer is an application's call to the TaskMan API to queue tasks. In this process the queuer defines the task and its environment. Applications are *not* allowed to do direct manipulation of the **^%ZTSCH** and **^%ZTSK** globals.

The TaskMan API consists of entry points that allow developers to create, manipulate, and inquire about tasks. The most widely used entry point, ^%ZTLOAD, lets developers queue tasks, which involves creating and scheduling them. First, an application sets the variables that ^%ZTLOAD needs to define the desired task. In turn, ^%ZTLOAD uses that information to create an entry in the TASKS (#14.4) file. ^%ZTLOAD then sets up a simple cross-reference to the new task in the SCHEDULE file, thereby finishing the queuing process.

After queuing the task, ^%ZTLOAD quits, returning control back to the queuer and leaving the next step in the process to the manager routines.

21.1.1.2 Option Scheduling through the OPTION SCHEDULING (#19.2) File

Another commonly used queuer is the OPTION SCHEDULING (#19.2) file. Menu Manager and TaskMan work together to allow certain options to be run as TaskMan tasks. These special options can be scheduled to run just once, or they can be set up to run over and over based on a rescheduling cycle. Such cycles can even include running the task whenever the computer system boots up.

21.1.2 Manager

For tasks to run, at least one CPU in a configuration needs to run a manager. Only one manager process needs to run per CPU; the site determines how many CPUs should be configured to run a manager. The manager's job is to route the tasks created by queuers. It normally runs at all times in the manager UCIs. It repeats the same loop of code all day long; every **2** seconds it looks for overdue tasks, every **15** seconds it checks the environment and performs some cleanup.

The environment check allows the system manager to control the manager even at its busiest. All of the commands to which the manager responds (described later) take effect here, between every task processed.

The manager looks for overdue tasks in the schedule list, comparing the current time to the start time of the tasks listed. If an overdue task is found, the manager removes it from the schedule list and inspects it. If the task is defined with a complete task record, the manager places it in a list of tasks ready to run. The manager places a task on one of several different lists depending on whether the task needs ownership of a currently unavailable **I/O** device. As its final step in processing each overdue task, the manager checks the number of submanagers available to process tasks and starts up new submanagers, if needed. The manager uses the **JOB** command (or **%SPAWN** if the manager is running in a DCL context on a Caché system).

The only variation on this scheme happens when the manager finds a task bound for a different Volume Set. Depending on the system configuration, such tasks may need to be run by the manager running on that other Volume Set. In this case, the current Volume Set's manager copies the task over to the Volume Set on which the task should run and marks it as moved in the current TASKS (#14.4) file. In this process, the task is assigned a new task number, and the manager on that other Volume Set handles the task from there. If during this process the manager discovers that the link between the two Volume Sets has dropped, it saves the task in a list of tasks waiting for that Volume Set and checks periodically to see whether it has been restored. When the link recovers, the manager sends, in sequence, all the waiting tasks to the other Volume Set.

The manager never actually runs the task but merely places it in a list as a task now available to be run by a submanager.

21.1.3 Submanagers

Submanagers are the processes that actually run tasks. A manager starts submanagers whenever more are needed to handle the current workload of tasks, and they only last as long as they are needed. Submanagers loop back and forth between finding new tasks to run and running them.

To run each task, the submanager first removes the task from the list of waiting tasks on which it reside (e.g., the Job or the **I/O** list). Then it looks up the task's entry in the TASKS (#14.4) file, unloading all of the information about the task. If the task needs a device, the submanager calls the Device Handler to get ownership of it and issues a **USE** command for it. Then the submanager sets up the partition for the task and does the following:

- Sets the priority.
- Cleans out unwanted variables.
- Sets up requested variables.
- Prints a page header on the device if one was requested, etc.

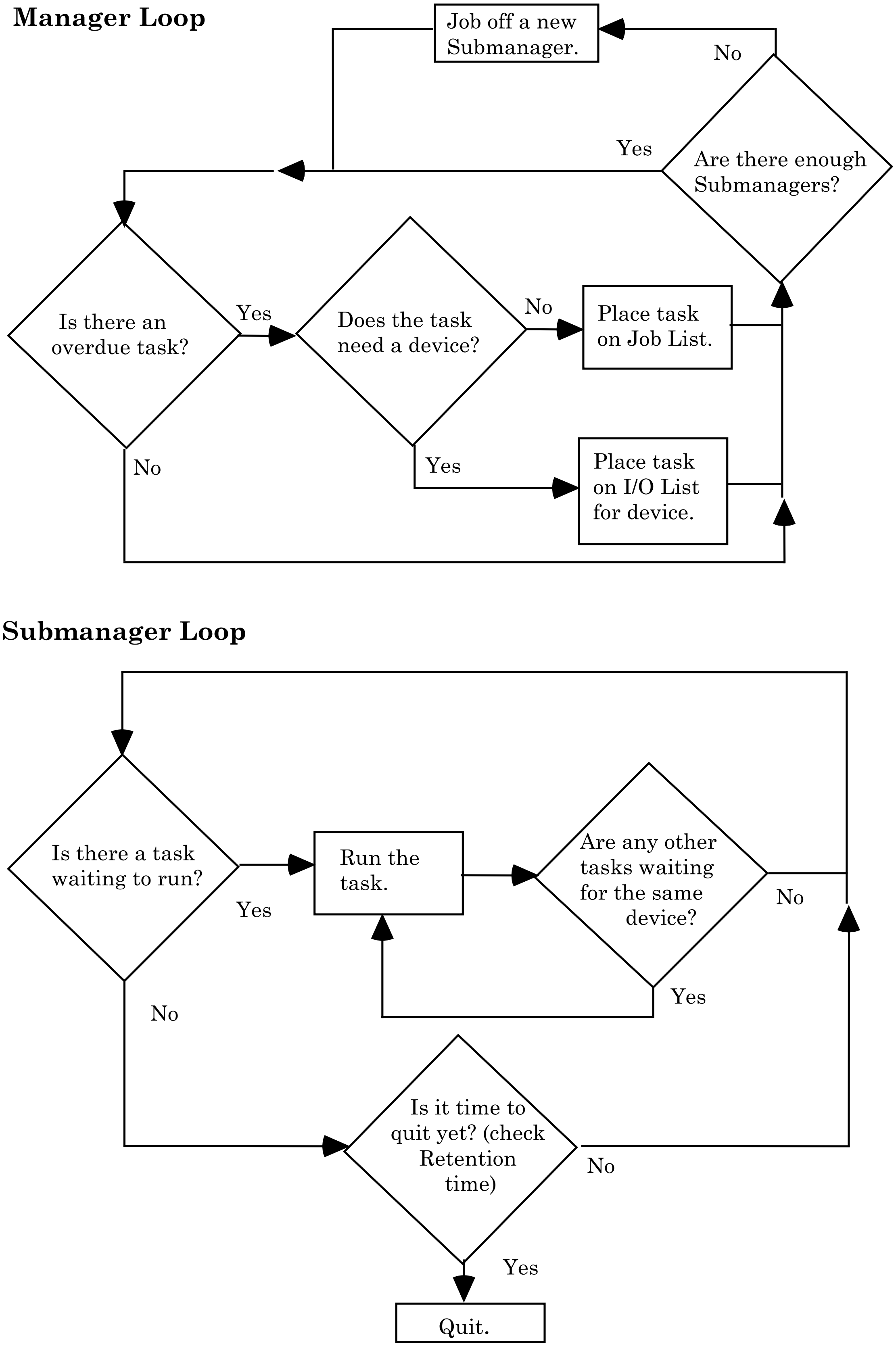
Next, the submanager starts the task running at the task's entry point. The submanager uses a **DO** command and runs the task's entry point in its own partition. When the task finishes, the submanager cleans up after the task:

- Closes the output device.
- Performs any commands left for it by the task, etc.

Running completely without user interaction, each task performs the work it was created to do and then quits, returning control to the submanager that started it. The task may leave instructions for its submanager, such as to requeue the task so that it runs again later or to delete the task's entry from the TASKS (#14.4) file, but the task itself finishes before the submanager continues.

After submanagers have run all available tasks, they wait an interval before quitting. This period, called submanager retention time, allows the submanager to keep its partition open for new tasks for a while so that the manager need *not* start a new submanager. Every time a new task shows up during the retention time, the submanager starts its main loop over again, returning to retention again only after all new tasks have been run. When the submanagers eventually reach the end of their retention time, they quit.

Figure 261: TaskMan Manager and Submanager Process Flow Diagram



Manager Loop

Job off a new Submanager.
No
Yes
Are there enough Submanagers?
Is there an overdue task?
Yes
Does the task need a device?
No
Place task on Job List.
No
Yes
Place task on I/O List for device.

Submanager Loop

Is there a task waiting to run?
Yes
Run the task.
Are any other tasks waiting for the same device?
No
Yes
No
Is it time to quit yet? (check Retention time)
No
Yes
Quit.

21.2 TaskMan's Files

The two central files that facilitate task processing are:

- TASKS (#14.4) file
- SCHEDULE file (*not* VA FileMan-compatible)

TaskMan is configured by three configuration Files:

- VOLUME SET (#14.5)
- UCI ASSOCIATION (#14.6)
- TASKMAN SITE PARAMETERS (#14.7)

These files and the TaskMan routines fall within TaskMan's namespace (**ZTM**) and numberspace. TaskMan user interface routines have been moved to the **XUTM** namespace beginning with Kernel 8.0 (they were previously in the **ZTM** namespace).

TaskMan also relies upon software components outside of its direct control. As an integral part of Kernel, TaskMan accesses several files controlled by other Kernel modules and calls many software entry points as a whole. TaskMan's main external relation, however, is with VistA software applications through the queuers and the tasks they use.

21.2.1 TaskMan Globals: ^%ZTSCH and ^%ZTSK

The **^%ZTSCH** global holds the SCHEDULE file, and the **^%ZTSK** global holds the TASKS (#14.4) file. Every environment controlled by a single manager needs each of these globals in its library UCI. **%** globals are used to make these files accessible to all the UCIs in that environment so a single manager's influence spans all of those UCIs. When the environment spans Volume Sets, **^%ZTSCH** and **^%ZTSK** are translated across the Volume Sets included. They are never replicated because TaskMan updates them so frequently.

The **^%ZTSK** global is mostly defined by VA FileMan (beginning with Kernel 8.0), but the **^%ZTSCH** is not. Historically these globals were *not* VA FileMan-compatible. Now, the inquire, search, and print capabilities of VA FileMan can be used to study the TASKS (#14.4) file. At present, all edit access to these globals is restricted to the TaskMan options that edit the tasks in various ways.

REF: For a description of the structure of **^%ZTSCH** and **^%ZTSK**, see the "Troubleshooting" section in the "TaskMan: System Management—Operation" section.

21.2.2 SCHEDULE File

The SCHEDULE file holds all of the lists and nodes that TaskMan uses to manage itself and to schedule tasks. Some of these lists are:

- Schedule List (or Time Queue)
- Waiting List (or IO Queue)
- Job List
- Compute Server Job List (or C List)
- Link List
- Status List
- Run Node
- TaskMan Error Log
- Error Screens

The SCHEDULE file's function is split between identifying the status of active tasks and of TaskMan itself.

REF: For more information on these lists, see the "TaskMan: System Management—Operation" section.

Most of the lists in the SCHEDULE file describe tasks, as follows:

- **Schedule List**—Sorts all scheduled tasks by time, according to when they are supposed to begin running.
- **Waiting List**—Stores each task whose running was delayed because its I/O device was busy.
- **Job List**—Holds those tasks that can begin running immediately.
- **Link List**—Stores tasks whose running is delayed because of a dropped link to another Volume Set.
- **Task List**—Describes all actively running tasks.
- **Compute Server Job List**—Describes all tasks waiting to start on a Compute Server (cross-CPU queuing).

The role of tracking the status of TaskMan itself is split between lists of information and individual nodes and flags. The Status List is where the manager keeps track of its current condition; it is a list because system administrators may choose to run more than one manager in the same TaskMan environment. The **RUN** node is a place where TaskMan stamps the current time; this node reveals when TaskMan stops running. The TaskMan Error Log is a simple list in which TaskMan stores each error that occurs either within TaskMan itself or within the tasks that it runs. The Error Screens are screens that can be established by system administrators to prevent the recording of certain errors.

These lists and nodes, as well as others *not* described here, are the primary data structures that TaskMan uses to schedule and run tasks.

21.2.3 TASKS (#14.4) File

The TASKS (#14.4) file, unlike the SCHEDULE file, contains the tasks themselves.

Every task run by TaskMan is described by an entry in the TASKS (#14.4) file. Each entry is subscripted by a unique internal number, and **^%ZTSK(-1)** always equals the number of the most recently created task. The lists and nodes in **^%ZTSCH** store the tasks' numbers that are scheduled to run. Each task's entry consists of a **^%ZTSK(task #, 0)** node that contains most of the essential information about the task, several decimal nodes (**.1**, **.2**, **.25**, and **.26**) that store the remainder of the critical information, and a number of storage nodes under **^%ZTSK(task#,.3)** that store the names and values of parameters that TaskMan creates for the task. Left unchecked, this file tends to grow.

REF: For a description of the various means of controlling this growth, see the "TaskMan: System Management—Operation" section.

21.2.4 Other Files

The TASKS (#14.4) and SCHEDULE files, taken together, describe all the information about tasks on the system. A few more files are needed, however, to describe everything about how tasks are managed on the system.

The following three files are stored in **^%ZIS**:

- VOLUME SET (#14.5) file—Describes the computer system's Volume Sets and how they are configured into TaskMan environments.
- UCI ASSOCIATION (#14.6) file—Lists all the UCIs on the system and which Volume Sets they belong to. In more complicated systems, it is also used to describe how the UCIs in different environments correspond with one another.
- TASKMAN SITE PARAMETERS (#14.7) file—Lets the system manager divide up the environments by both CPU and Volume Set. This allows a fine degree of control over such parameters as priority, partition size, and retention time.

Taken together, these files give system administrators precise and powerful control over TaskMan's behavior.

Other minor pieces of information are scattered throughout other Kernel files, especially the DEVICE (#3.5) and OPTION SCHEDULING (#19.2) files.

21.3 System Configuration Terminology

TaskMan operates close to the level of the system architecture. It *must* be capable of starting tasks in all the environments within a computer system. This means it *must* know about those environments; consequently, the options, routines, files, and documentation somehow *must* refer to that architecture.

One problem presented by system configuration is terminology. Such system architecture features as UCIs, directories, Volume Sets, and namespaces are *not* part of the ANSI M standard, so different vendors use different terminology. Although it would be ideal for Kernel to use a universal terminology, none exists. For historical reasons, Kernel has settled on a terminology based on that of **DSM-11** that includes the terms in Table 40:

Table 40: TaskMan System Configuration Terminology

Term	Definition
UCI	User Class Identifier. This is roughly equivalent to a "directory" or an "account". A UCI refers to the environment limited to a particular set of routines and globals. In Caché terms, this is a "namespace."
Manager UCI	Roughly equivalent to a "system UCI" or a "library UCI." This is where the vendor's system management routines are kept and where all **%-**namespaced routines and globals reside. Currently, all Kernel **%** routines and globals are mapped back to the Production account.
Volume Set	On current systems, you just set this to the string "**ROU**". This is the critical definition, since this is what affects how TaskMan starts background jobs.
CPU	Also known as a "node" or "computer", this designates a source of computing power and partitions. It is used both for controlling TaskMan's behavior with parameters and for sending tasks to specific CPUs.
Mounted Volume Set	Obsolete; no longer used.

NOTE: The TaskMan topics in this section make use of this terminology.

21.4 TaskMan Security Key

The TaskMan module comes with one security key, ZTMQ. The ZTMQ security key does *not* completely lock any options. Instead, it affects the behavior of the following three options:

- **Dequeue Tasks** [XUTM DQ]
- **Requeue Tasks** [XUTM REQ]
- **Delete Tasks** [XUTM DEL]

Those who use these options without holding this security key can manipulate only their own tasks. Only the holder of the ZTMQ security key can use these options to manipulate any task on the system.

22 TaskMan: System Management—Configuration

This section discusses the many issues surrounding the configuration of TaskMan.

22.1 Defining TaskMan Environments

The part of configuring TaskMan for a system that requires the most creativity is deciding how to divide the system's UCIs, Volume Sets, and CPUs into TaskMan environments. A TaskMan environment is the collection of UCIs from which entries can be made directly into a given manager's TASKS (#14.4) and SCHEDULE files and that are within that manager's reach. This requires looking at the system in terms of queuing and starting tasks. There are a number of options available. Many different configurations are possible.

One type of configuration has CPUs sharing the same Volume Set. Since this type of environment shares a single Volume Set among multiple CPUs, they also share a single TASKS (#14.4) and SCHEDULE file. However, the reach of managers *cannot* span CPUs. Therefore, you *must* decide which CPUs in that environment run managers, or whether some of them should rely on the other CPUs to run their tasks for them. Alpha clusters in VA are typically configured with managers on only one or a few CPUs.

A different configuration allows you to limit the number of places TaskMan runs. In this scenario, you pick certain CPUs to run TaskMan and give them managers and files to do the job. To have background processing support, the remaining Volume Sets need to be able to queue to one of the managers on the system. This entails translating the TASKS (#14.4) and SCHEDULE files of that manager so they are visible to the unsupported Volume Set. To tell TaskMan that the one Volume Set runs no tasks but is instead supported by the other, you *must* configure the VOLUME SET (#14.5) file as described later in this section.

Another possible configuration is to allow tasks to run everywhere, which requires that you place managers within reach of every UCI and that you define your TaskMan environments accordingly. Under this configuration every CPU needs its own manager, and its own TASKS (#14.4) and SCHEDULE files.

One other configuration to keep in mind, of course, is to have a standalone environment disconnected from the rest of the computer system. Such environments make excellent test areas for developers. They are configured the same regardless of the configuration of the main system.

22.2 Configuring TaskMan

TaskMan's three configuration files *must* be setup to properly reflect your system's layout. The three files are:

- TASKMAN SITE PARAMETERS (#14.7)
- VOLUME SET (#14.5)
- UCI ASSOCIATION (#14.6)

22.2.1 Edit TaskMan Parameters Menu

The following three options on the **Edit TaskMan Parameters** [XUTM PARAMETER EDIT] menu allow you to edit each of the three configuration files:

- **Site Parameters Edit** [XUTM BVPAIR]

  REF: For more information on the **Site Parameters Edit** [XUTM BVPAIR] option and the TASKMAN SITE PARAMETERS (#14.7) file, see the "TASKMAN SITE PARAMETERS (#14.7) File" section.

- **UCI Association Table Edit** [XUTM UCI]

  REF: For more information on the **UCI Association Table Edit** [XUTM UCI] option and the VOLUME SET (#14.5) file, see the "UCI ASSOCIATION (#14.6) File" section.

- **Volume Set Edit** [XUTM VOLUME]

  REF: For more information on the **Volume Set Edit** [XUTM VOLUME] option and the UCI ASSOCIATION (#14.6) file, see the "VOLUME SET (#14.5) File" section.

Because the TASKMAN SITE PARAMETERS (#14.7) allows you to define parameters (e.g., TaskMan Job Limit) separately for each CPU on your system; you are able to optimize TaskMan's behavior individually for each CPU.

You no longer need to stop and then restart TaskMan in order to change the TASKMAN JOB LIMIT on a CPU. Cross-references on the relevant fields locate every TaskMan on your system and inform them that they need to update their TaskMan parameter information. Thus, within a minute or so of making the changes, TaskMan on that CPU should be operating with the new value.

22.2.2 TaskMan's Reach

The key issue that defines TaskMan's configuration is its "reach," those places where TaskMan can start background jobs. TaskMan's reach extends to:

- All UCIs a submanager can access directly after using Kernel's UCI switching facilities.
- All other managers TASKS (#14.4) and SCHEDULE files to which a given manager can **WRITE** using extended global reference.
- All UCIs on Print Servers with link access to the current Volume Set.

TaskMan's reach does *not* include other sites on a wide area network, because they *cannot* be accessed through either UCI switching or through extended global reference. There are ways to simulate such a reach through the use of server options, however. For purposes of TaskMan configuration, you generally think in terms of the reach of a single manager, which can only run tasks in the UCIs it can reach.

22.2.3 TASKMAN SITE PARAMETERS (#14.7) File

Figure 262: Site Parameters Edit Option

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Task Manager ...                                                   [XUTM MGR]
   Taskman Management Utilities ...                               [XUTM UTIL]
      Edit Taskman Parameters ...                       [XUTM PARAMETER EDIT]
         Site Parameters Edit                                   [XUTM BVPAIR]
```

System managers *must* enter one set of site parameters into the TASKMAN SITE PARAMETERS (#14.7) file for each manager that runs in a different Volume Set/CPU.

Table 41 lists the set of parameters that tells each manager how it should process tasks. The parameters are organized both by Volume Set and by CPU. This allows two CPUs that share a Volume Set to be treated differently if one is more powerful than the other.

Table 41: TASKMAN SITE PARAMETERS (#14.7) File—Field Entries

Field	Description
BOX-VOLUME PAIR (#.01)	The BOX-VOLUME PAIR field identifies a Volume Set and the CPU on which it is available. It contains the name of a Volume Set concatenated to the CPU ("box") name: first the Volume Set name and then the CPU name. For example, if the Volume Set name is "**KRN**" and the name of the CPU (e.g., box) is "**ABC999**," then the BOX-VOLUME PAIR would be "**KRN:ABC999**." For systems on which each CPU tends to have a unique Volume Set and vice versa, you can enter just the Volume Set name (e.g., "PSA" or "AAA"). This field's value for the current process can be found by doing GETENV^%ZOSV and checking the fourth **^**-piece of Y. Since the Volume Set and CPU are identified, the TaskMan site parameters can be tuned for each specific Volume Set and CPU affected. Systems running managers on more than one CPU need one entry for each CPU where a manager is running.
LOG TASKS? (#2)	Set the LOG TASKS? field to **YES** to make tasks log in and out through the signon log the way

Field	Description
	interactive users do. How to set this is up to the individual site; it does consume space and resources.
TASK PARTITION SIZE (#4)	The TASK PARTITION SIZE field is used to assign partition sizes for tasks. The value from this field is plugged directly into the **JOB** command used to create new submanagers. If this field is left blank, all tasks receive the operating system's current default value. This field should only be used by system managers who thoroughly understand how their vendor's version of M handles partition sizes with the **JOB** command.
SUBMANAGER RETENTION TIME (#5)	The SUBMANAGER RETENTION TIME number determines how many seconds submanagers should wait while looking for new tasks. The purpose of this field is to reduce the number of **JOB** commands needed to process a site's tasks. By keeping old submanagers around to run new tasks, new process creation is significantly reduced.
TASKMAN JOB LIMIT (#6)	If there are more active processes on the system than the number stored in the TASKMAN JOB LIMIT field, TaskMan does *not* create new submanagers to handle tasks. Task processing is left to existing submanagers until the number of processes falls back below this number. This number should be slightly lower than the MAX SIGNON ALLOWED (#41,2) field of the VOLUME SET (#41) Multiple field in the KERNEL SYSTEM PARAMETERS (#8989.3) file so that the system manager still has room to sign on when TaskMan is using its greatest number of partitions.
TASKMAN HANG BETWEEN NEW JOBS (#7)	The TASKMAN HANG BETWEEN NEW JOBS field sets a delay between the creation of new submanagers, in seconds. It is useful as a throttle. For systems, this delay spaces out the use of the **JOB** command to avoid slowing users' response time when the manager needs to JOB off many new processes in rapid succession. For systems that create new processes cheaply, this delay is unnecessary. This delay also becomes less important when a high submanager retention time is used since higher retention times reduce the likelihood that TaskMan needs to create new processes. Be sure *not* to combine a high TASKMAN HANG BETWEEN NEW JOBS value with a low

Field	Description
	SUBMANAGER RETENTION TIME value, since that increases the number of jobs per day TaskMan has to start and can cause busy systems to fall behind. The number should be the lowest value that prevents the problem and can be left blank for systems with efficient **JOB** commands.
MODE OF TASKMAN (#8)	The MODE OF TASKMAN field determines how each CPU (BOX-VOLUME pair entry) should process tasks. You can set it to one of four values: • **General Processor ("G"):** The **G** type should be selected when the TASKS (#14.4) and Scheduling files are seen by only one Volume Set. For example, VA's Alpha clusters have several CPUs, but each of them runs on the same Volume Set. The manager on a **G** type runs tasks created on the same Volume Set and tasks from any other Volume Set that explicitly requests the **G** type's Volume Set. The **G** type sends tasks from another Volume Set that did *not* explicitly request its Volume Set back to the originating Volume Set, however. To transfer tasks to a **G** type, TaskMan uses extended global references to copy the task to the destination TASKS (#14.4) and Scheduling files and then removes the task from its own side. Submanagers started on a **G**-type processor process tasks in the Partition Waiting List and the Busy Device Waiting List. • **Print Server ("P"):** The **P** type should be selected when multiple Volume Sets map to the same TASKS (#14.4) and Scheduling files, and you want to run the manager on the Volume Set/CPU in question. Like the **G** type, the manager on a **P** type runs tasks created on the same Volume Set and tasks from any other Volume Set/CPU that explicitly request the **P** type's Volume Set/CPU. Unlike the **G** type, however, the **P** type also runs tasks from other Volume Sets that did *not* make an explicit Volume Set request. Tasks are transferred to a **P** type in the same way as to a **G** type, and submanagers behave the same. • **Compute Server ("C"):** The **C** type should be selected when multiple Volume Sets map to the same TASKS (#14.4) and Scheduling files (as

Field	Description
	with the **P** type), but when the Volume Set/CPU in question runs users (*not* tasks). The manager does *not* start on a **C** type. Tasks that explicitly request to run on a **C** type are transferred to it by being placed in the Link Waiting List; a submanager is then jobbed across to the **C** type Volume Set/CPU. Submanagers started on a **C** type only process tasks in the Link Waiting List for their Volume Set. • **Other Non-TaskMan ("O"):** Neither the manager nor the submanager runs on **O** types. Tasks sent from or to an **O** type are rejected. Because of the field's crucial role in guiding TaskMan's behavior, the field is required.
VAX ENVIRONMENT FOR DCL (#9)	The VAX ENVIRONMENT FOR DCL field only has meaning to DSM for OpenVMS and Caché systems. It is set to the OpenVMS username of the DSM environment manager account. Setting it to this username causes the manager to use **%SPAWN** to SUBMIT submanagers to run. This method requires that certain DCL command files exist, along with a TASKMAN OpenVMS user account and directory. **REF:** For descriptions of the needed setups, see the "Running TaskMan with a DCL Context" section. If the field is empty, the manager starts submanagers with the **JOB** command instead.
LOAD BALANCE ROUTINE (#21)	If you are running multiple managers (one per node), use the LOAD BALANCE ROUTINE field to set up load balancing between the managers on each node. It should be set to the name of an extrinsic function that returns a load rating for the node. **REF:** For more information on load balancing, see the "Multiple TaskMan Managers and Load Balancing" section.

22.2.4 VOLUME SET (#14.5) File

Figure 263: Volume Set Edit Option

```
SYSTEMS MANAGER MENU ...                                              [EVE]
Task Manager ...                                                 [XUTM MGR]
   Taskman Management Utilities ...                             [XUTM UTIL]
      Edit Taskman Parameters ...                     [XUTM PARAMETER EDIT]
         Volume Set Edit                                      [XUTM VOLUME]
```

TaskMan knows about a system's configuration from the values entered into the VOLUME SET (#14.5) file using the **Volume Set Edit** [XUTM VOLUME] option. The information stored in this file strongly affects TaskMan's behavior. If you inaccurately describe your system, you usually notice very quickly as TaskMan begins processing tasks in a consistently incorrect way.

You need to make one entry in this file for each Volume Set that tasks can be queued to or from. These entries are only used when:

- A manager is running on the Volume Set and *must* look up information about its own environment.
- The Volume Set is a required volume, in which case every manager *must* check access to it when they start up.
- A task needs to run on the Volume Set, in which case the manager *must* look up how to get the task there.

Figure 264 shows what is set up for FORUM:

Figure 264: Sample Volume Set Setup on FORUM

```
VOLUME SET (14.5)

VOLUME SET: ROU                         INHIBIT LOGONS?: NO
  LINK ACCESS?: NO                      TASKMAN FILES UCI: VAH
  DAYS TO KEEP OLD TASKS: 1             TYPE: GENERAL PURPOSE VOLUME SET
  SIGNON/PRODUCTION VOLUME SET: Yes

UCI ASSOCIATION (14.6)

Empty


TASKMAN SITE PARAMETERS (14.7 )

BOX-VOLUME PAIR: ROU:FORFORUM1          LOG TASKS?: NO
  SUBMANAGER RETENTION TIME: 60         TASKMAN JOB LIMIT: 400
  TASKMAN HANG BETWEEN NEW JOBS: 1      MODE OF TASKMAN: GENERAL PROCESSOR
  OUT OF SERVICE: NO                    MIN SUBMANAGER CNT: 10
  LOAD BALANCE ROUTINE: $$CACHE1()      Auto Delete Tasks: Yes
  Manager Startup Delay: 30
```

The value of **^%ZOSF(“VOL”)** is **FOR**.

Table 42: VOLUME SET (#14.5) File—Field Entries

Field	Description
VOLUME SET (#.01)	The VOLUME SET field should be set to the name of a Volume Set. It is used in extended global references to reach this Volume Set and can be used in UCI-switching software to move submanagers between UCIs. If you are unsure how your Volume Sets are named, you can look at the value of **^%ZOSF(“VOL”)** in the Volume Set in question.
TYPE (#.1)	The TYPE field is used to help resolve where tasks should run; it should properly identify the type of the Volume Set. Typically it should be set to the same value as the MODE OF TASKMAN (#8) field for all BOX-VOLUME PAIRs associated with this Volume Set, in the TASKMAN SITE PARAMETERS (#14.7) file. This field *must* be filled in for all Volume Sets. This field can have the following values: • **G—**GENERAL PURPOSE VOLUME SET • **P—**PRINT SERVER • **C—**COMPUTE SERVER • **O—**OTHER NON-TASKMAN VOLUME SET These values have the same meanings as the equivalent values for the MODE OF TASKMAN (#8) field in the TASKMAN SITE PARAMETERS (#14.7) file, as described previously in the “TASKMAN SITE PARAMETERS (#14.7) File” section. GENERAL PURPOSE VOLUME SET for Volume Sets is the rough equivalent of the MODE OF TASKMAN value GENERAL PROCESSOR for BOX-VOLUME PAIRs. **NOTE:** The **FILE SERVER** value has been removed; Volume Sets for File Servers should be set to a TYPE of **OTHER NON-TASKMAN VOLUME SET**.
INHIBIT LOGONS? (#1)	Setting the INHIBIT LOGONS? field to **YES** causes TaskMan to notify Signon that logons are now prohibited and to enter a **PAUSE** state (stopping processing of tasks) until logons are allowed again.

Field	Description
	Under ordinary circumstances, system managers should leave this field as **NULL** or **NO**.
LINK ACCESS (#2)	The LINK ACCESS field should always be set to **NULL** or **YES** for the usual kinds of configurations used in VistA. Answer **NO** to tell TaskMan that this Volume Set *cannot* be accessed by other Volume Sets using the local network links. Tasks that request a Volume Set without link access are rejected by TaskMan. Such Volume Sets are usually PC workstations linked into the larger network. They can access the core computers but *cannot* be accessed themselves. Some system managers may wish to have a completely isolated computer for testing. They can cut it off from the rest of the world by making entries for all the other Volume Sets and setting this field to **NO** for each of them. This explicitly tells TaskMan it *cannot* reach the other Volume Sets.
OUT OF SERVICE? (#3, Obsolete, see TYPE field)	The OUT OF SERVICE? field is obsolete and should only be set to **NULL**; use the TYPE (#.1) field.
REQUIRED VOLUME SET? (#4, Obsolete)	The REQUIRED VOLUME SET? field is obsolete and should only be set to **NULL**.
TASKMAN FILES UCI (#5)	The TASKMAN FILES UCI field should be set to the name of the UCI that holds the **^%ZTSCH** and **^%ZTSK** globals (usually the manager UCI). The answer should *not* contain a comma and Volume Set name (e.g., **VAH,PSA**), just the UCI name (e.g., **VAH**). This field is required.
TASKMAN FILES VOLUME SET (#6)	The TASKMAN FILES VOLUME SET field should be set to the name of the Volume Set that holds **^%ZTSCH** and **^%ZTSK**. A **NULL** value means this Volume Set holds its own TaskMan files, which is usually the case.
REPLACEMENT VOLUME SET (#7)	The REPLACEMENT VOLUME SET field should be set to the name of a Volume Set to which tasks can be sent if this Volume Set is unavailable. A REPLACEMENT VOLUME SET should be essentially equivalent in features to the current one, since tasks that would normally run on the current one are running on the REPLACEMENT VOLUME SET instead. For many Volume Sets, no other Volume Set is equivalent, and tasks should wait for the link to be restored rather than run elsewhere. If

Field	Description
	tasks that need this Volume Set should wait, leave the field blank.
DAYS TO KEEP OLD TASKS (#8)	The number stored in the DAYS TO KEEP OLD TASKS field is used by the **Queuable Task Log Cleanup** [XUTM QCLEAN] option to decide which tasks to delete. The decision only affects inactive tasks, as explained in the discussion of the **Queuable Task Log Cleanup** [XUTM QCLEAN] option. Values in this field *cannot* inadvertently cause TaskMan to delete scheduled or running tasks. If the field contains no value, **XUTM QCLEAN** keeps the last seven days' tasks. A value of **0** here keeps your file very clean.

22.2.5 UCI ASSOCIATION (#14.6) File

Figure 265: UCI Association Table Edit Option

```
SYSTEMS MANAGER MENU ...                                        [EVE]
Task Manager ...                                            [XUTM MGR]
   Taskman Management Utilities ...                        [XUTM UTIL]
      Edit Taskman Parameters ...                [XUTM PARAMETER EDIT]
         UCI Association Table Edit                         [XUTM UCI]
```

There are two different kinds of entries made into the UCI ASSOCIATION (#14.6) file using the **UCI Association Table Edit** [XUTM UCI] option:

- Partial File Entries
- Complete File Entries

22.2.5.1 Partial File Entries

File entries with the following first two fields filled in identify the valid UCIs on the system for TaskMan:

- FROM UCI (Table 43)
- FROM VOLUME SET (Table 43)

Every VistA site needs one entry of this type for each UCI to which tasks can be queued or from which tasks are created.

NOTE: Caché sites only need to fill in these first two fields.

REF: For a sample configuration, see the "Sample Configuration: Standardized VA Caché and GT.M Configuration" section.

22.2.5.2 Complete File Entries

File entries with all four fields (Table 43) completed collectively build a UCI ASSOCIATION TABLE.

A complete UCI ASSOCIATION TABLE tells TaskMan which UCI to use for tasks that *must* switch Volume Sets in order to reach an I/O device. This situation arises when an I/O device is located in a different Volume Set than the Volume Set where the task was created. In such situations, the manager knows exactly where the task originated and knows to which Volume Set it *must* be moved, but it does *not* know in which UCI on that Volume Set it should run the task. A UCI ASSOCIATION TABLE entry supplies the missing information by linking equivalent UCIs together. When building a full UCI ASSOCIATION TABLE, you can omit entries where the UCIs on both Volume Sets have the same name because TaskMan assumes that same-named UCIs are equivalent if no entry is present.

Table 43: UCI ASSOCIATION (#14.6) File—Partial and Complete Field Entries

Field	Description
FROM UCI (#.01)	The FROM UCI field should be set to the name of a UCI on your system. Enter only the UCI name (e.g., **VAH**). Do *not* include the Volume Set name (e.g., **VAH,ROU**). • For entries requiring only two fields, this catalogues all the UCIs on your system (and there should be an entry for each). • For four-field entries, this represents a UCI from which tasks are being transferred in order to reach their **I/O** device.
FROM VOLUME SET (#1)	The FROM VOLUME SET field should be set to the name of the Volume Set that holds the UCI identified in the entry's FROM UCI (#.01) field. Every Volume Set listed in this field should be described in the VOLUME SET (#14.5) file. • For four-field entries, this represents the Volume Set from which tasks are being transferred in order to reach their **I/O** device.
TO VOLUME SET (#2)	The TO VOLUME SET field is only used for entries that build a UCI Association Table. For such entries, it should be the name of the Volume Set to which tasks are being transferred in order to reach their **I/O** devices.
TO UC (#3)I	As with TO VOLUME SET(#2), the TO UCI field is only used for entries that build a UCI Association Table. For such entries, it should be the name of the UCI to which tasks are transferred whenever they *must* be moved from the UCI on the first Volume Set to the second Volume Set in order to reach their **I/O** devices.

Field	Description
	As with the From UCI field, the Volume Set name should *not* be included.

22.2.6 Sample Configuration: Standardized VA Caché and GT.M Configuration

Sites that run managers on their satellites should make the appropriate TASKMAN SITE PARAMETERS (#14.7) file entries for each satellite and adjust their TaskMan Job Limit to reflect each satellite’s individual capacity.

Figure 266: VOLUME SET (#14.5) File Standardized VA Caché and GT.M Configuration

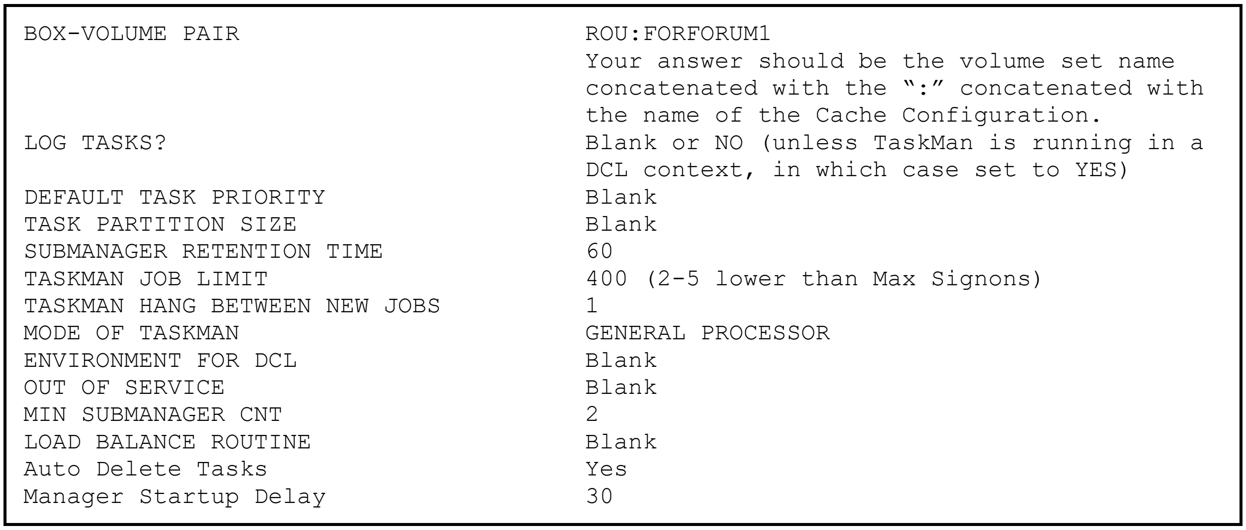
```
VOLUME SET                          You need one entry, for ROU
TYPE                                GENERAL PURPOSE VOLUME SET
INHIBIT LOGONS?                     Blank or NO
LINK ACCESS?                        Blank or NO
OUT OF SERVICE?                     Blank or NO
REQUIRED VOLUME SET?                Blank or NO
TASKMAN FILES UCI                   VAH
TASKMAN FILES VOLUME SET            Leave this blank
REPLACEMENT VOLUME SET              Leave this blank
DAYS TO KEEP OLD TASKS              Up to you; can leave blank
SIGNON/PRODUCTION VOLUME SET        Yes
```

Figure 267: UCI ASSOCIATION (#14.6) File—Standardized VA Caché and GT.M Configuration

```
FROM UCI                            1 entries: VAH
FROM VOLUME SET                     ROU
TO VOLUME SET                       Blank
TO UCI                              Blank
```

NOTE: You can leave this empty.

Figure 268: TASKMAN SITE PARAMETERS (#14.7) File Standardized VA Caché and GT.M Configuration



```
BOX-VOLUME PAIR                        ROU:FORFORUM1
                                       Your answer should be the volume set name
                                       concatenated with the ":" concatenated with
                                       the name of the Cache Configuration.
LOG TASKS?                             Blank or NO (unless TaskMan is running in a
                                       DCL context, in which case set to YES)
DEFAULT TASK PRIORITY                  Blank
TASK PARTITION SIZE                    Blank
SUBMANAGER RETENTION TIME              60
TASKMAN JOB LIMIT                      400 (2-5 lower than Max Signons)
TASKMAN HANG BETWEEN NEW JOBS          1
MODE OF TASKMAN                        GENERAL PROCESSOR
ENVIRONMENT FOR DCL                    Blank
OUT OF SERVICE                         Blank
MIN SUBMANAGER CNT                     2
LOAD BALANCE ROUTINE                   Blank
Auto Delete Tasks                      Yes
Manager Startup Delay                  30
```

22.3 Manager Startup

You may want to configure your system so that, on CPUs where the manager should run, a manager starts up every time the CPU starts up. Otherwise, you need to manually start up the manager each time you start up those nodes that should run the manager.

For most sites, only one manager is needed to cover each environment. Therefore, this section focuses on starting up only a single manager.

Neither the manager nor the submanagers starts up on a BOX-VOLUME PAIR pair of the wrong type, so pay attention to how you fill in the MODE OF TASKMAN field of the TASKMAN SITE PARAMETERS (#14.7) file. If you want the manager to start, you *must* make sure this field is set to either a Print Server or a General Processor.

Getting the manager to start up when the system does is accomplished in the VA by the **ZSTU** routine in the **%SYS** namespace. This routine is provided by Enterprise Product Support (EPS).

22.4 Multiple TaskMan Managers and Load Balancing

TaskMan supports the running of multiple manager processes; however, only one manager process should run per CPU. Running multiple managers is probably useful only at large sites; at a large site, doing this can enable tasks to be processed more quickly than if only one CPU runs a manager. An added bonus with multiple managers is that if one CPU running a manager becomes unavailable, managers still run on the other CPUs, with no further re-configuration required.

22.4.1 Configuration for Multiple Managers

Each node that runs a TaskMan manager *must* have its own entry (BOX-VOLUME PAIR) in the TASKMAN SITE PARAMETERS (#14.7) file.

Each CPU *must* share access to a common **^%ZTSK** and **^%ZTSCH** global and have access to the same devices. Because of this, all CPUs *must* run the same M implementation.

22.4.2 Starting Up, Pausing, and Stopping Multiple Managers

You need to start a manager on each CPU where a manager should run. Whatever steps you follow to start a single manager, you need to repeat for any additional nodes on which you want to run additional managers.

The options that place TaskMan in a **WAIT** state and stop TaskMan are *not* CPU-specific; they affect all running managers across the system.

22.4.3 Load Balancing

The LOAD BALANCE ROUTINE field in the TASKMAN SITE PARAMETERS (#14.7) file holds the name of a function that returns a CPU's load rating. This field is only useful if you are running multiple TaskMan managers.

To use load balancing, enter a routine name in the LOAD BALANCE ROUTINE field for each participating CPU's BOX-VOLUME PAIR entry. Kernel patch XU*8.0*355 added the following routine for TaskMan load balancing in Caché:

> **$$CACHE2(@com-file,logical-name)** in **^ZTM6**

If the com-file value is set, that com-file runs each time TaskMan gets the balance value. The logical-name defaults to **VISTA$METRIC** or uses the value entered. The normal way would be to have **$$CACHE2()** in the field and use the following two scripts:

- **GET_METRIC.COM**—This script sets the logical "**VISTA$METRIC**." It can be run by TaskMan or from the **TM$<*node*>** batch queue with the **METRIC_SCHEDULE.COM** script.
- **METRIC_SCHEDULE.COM**—This script takes a parameter of the number of seconds to reschedule itself. It defaults to **15** seconds and runs under the "**SYSTEM**" user.

NOTE: These scripts are located in the same directory as the TaskMan in DCL files.

Use of TaskMan in DCL is optional.

It is all right to run multiple TaskMan managers without using load balancing; it is also all right if load balancing is set up and only one manager is running (that manager automatically takes all jobs itself). If one manager's CPU has the LOAD BALANCE ROUTINE field filled in and another running manager's CPU does not, the managers acts as if no load balancing is taking place. In short, the only ramification from various combinations of managers with the LOAD BALANCE ROUTINE field filled in or not is that load balancing might *not* take place.

The load balancing routine *must* be an extrinsic function that returns a positive value. The CPU with the highest value is the one that runs new tasks.

Cache Algorithms:

- **$$Cache2()**—Returns the TCPIP metric.
- **$$Cache1()**—Returns the Available jobs.

Each CPU performing load balancing compares its current CPU capacity with that of the other nodes running managers. If the current CPU has a lower rating than the other CPUs, it puts itself in a **BALANCE** state and waits to let the other CPUs take up the load before running more jobs itself.

Submanagers try and wait until there node is running before testing if they should exit.

22.4.4 Monitor Taskman Option

On a system where multiple managers are running, the **Monitor Taskman** [XUTM ZTMON] option shows a combined view of the operation of multiple managers.

If the current node (the one where you are running the **Monitor Taskman** [XUTM ZTMON] option) has a lower rating than other nodes, Monitor TaskMan shows that the current node is in a **BALANCE** state.

22.5 Device Handler’s Influence on TaskMan

Certain DEVICE (#3.5) file fields strongly affect TaskMan’s behavior. System managers should keep these effects in mind as they configure their systems’ devices.

Table 44: DEVICE (#3.5) file—TaskMan-related Field Entries

Field	Description
VOLUME SET(CPU) (#1.9)	If the VOLUME SET(CPU) field is *not* filled in, TaskMan considers this device to be available from all Volume Sets. If it is filled in, TaskMan makes sure all tasks that need this device start on the designated Volume Set.
TYPE (#2)	Any tasks that *must* wait for HFS- or SPL-type devices are rescheduled for **ten minutes** in the future, instead of being placed in a list of waiting tasks. This is because these lists are checked through repeated opens, which may contaminate the output of these two special types of devices.
PRIORITY AT RUN TIME (#25)	The PRIORITY AT RUN TIME field overrides the default priority that system managers can establish for tasks using the **Site Parameters Edit** [XUTM BVPAIR] option on the **Edit TaskMan Parameters** [XUTM PARAMETER EDIT] menu.
TASKMAN PRINT A HEADER PAGE? (#26)	If the TASKMAN PRINT A HEADER PAGE? field is set to **YES** for the device being opened by the submanager, a header page is printed. The header page distributed with TaskMan is very simple, and system managers can substitute their own locally written header pages. To do this, you *must* rename your header page routine as **^%ZTMSH**, the name of the one distributed with TaskMan. Whenever you install new versions of Kernel, it overwrites **^%ZTMSH** with the default copy, so you should maintain your local version by doing the following: • Keep your local header page routine saved somewhere under a local name. • After each Kernel install, re-save the locally named copy as **^%ZTMSH**.

Figure 269 shows an alternative to the default header page distributed with Kernel:

Figure 269: Customized Header Page Routine

```
%ZZTMSH      ;SEA/RDS-Local: Sample Header Page ;3/9/92 11:17 ;
             ;;1.0;Local;;
             ;
LOCAL        ;Print The Local Header Page
             ;
B            ;build text lines
             S X1=$P($G(^VA(200,DUZ,0)),U) I X1=“” S X1=“name unknown”
             S X2=$P($G(^VA(200,DUZ,5)),U,2) I X2=“” S X2=“unlisted mail stop”
             S X3=$P($G(^VA(200,DUZ,.13)),U,2) I X3=“” S X3=“unlisted phone number”
             S ZZLINE1=$$FORMAT(“ “_X1_”  (“_X2_”)  “_X3_””,IOM)
             S ZZLINE2=$$FORMAT(“ “_ZTDESC_” “,IOM)
             S ZZLINE3=$$FORMAT(“ “_ION_”  “_$$HTE^XLFDT($H)_””,IOM)
             ;
D            ;display each line three times
             F X=1:1:3 W !,ZZLINE1
             W ! F X=1:1:3 W !,ZZLINE2
             W ! F X=1:1:3 W !,ZZLINE3
             Q
             ;
FORMAT(ZZTEXT,ZZIOM)  ;local extrinsic function
             ;input: text to be formatted, and margin width
             ;output: text filled out to margin width -3 with *characters
             N ZZ1,ZZFILLED
             S ZZ1=ZZIOM-3-$L(ZZTEXT)\2
             S $P(ZZFILLED,”*”,ZZ1*2+1)=““
             S $P(ZZFILLED,”*”,ZZ1+1)=ZZTEXT
             I $L(ZZFILLED)+3-ZZIOM S ZZFILLED=ZZFILLED_”*”
             Q ZZFILLED
```

Figure 270: Customized Header Page

```
    ************ XUUSER,ONE  (OIFO)  FTS 555-5555 ************
    ************ XUUSER,ONE  (OIFO)  FTS 555-5555 ************
    ************ XUUSER,ONE  (OIFO)  FTS 555-5555 ************

    ********************** SAMPLE TASK ***********************
    ********************** SAMPLE TASK ***********************
    ********************** SAMPLE TASK ***********************

    *********** LAT DEVICE  Jun 30, 1992@14:34:01 ************
    *********** LAT DEVICE  Jun 30, 1992@14:34:01 ************
    *********** LAT DEVICE  Jun 30, 1992@14:34:01 ************
```

22.6 Running TaskMan with a DCL Context

When run from a DCL context, TaskMan runs as an OpenVMS user. The manager runs as a job that originates from a node-specific OpenVMS batch queue and, by default, submits new submanagers to the same queue as needed.

One advantage to running TaskMan from a DCL context is that it allows jobs to be queued to specific CPUs. When a program calls ^%ZTLOAD, it can request that the job run on a specific CPU/node in your cluster (via the **ZTCPU** input variable). Unless you are running TaskMan in a

DCL context (on Caché systems only), this request will probably fail (and possibly cause the task *not* to run). When TaskMan runs with a DCL context, however, the manager can submit the job as a new submanager to a given CPU's TaskMan batch queue.

Depending on the **%ZTSK** and **%ZTSCH** mapping, multiple Cache environments on the same CPU can each run TaskMan in a DCL context. Although TaskMan in each Cache environment shares the same account, directory, DCL command files, and batch queue, jobs run in the environment specified in each environment's VAX ENVIRONMENT FOR DCL site parameter.

NOTE: Kernel patch XU*8.0*355 added the **$$CACHE2** routine for TaskMan load balancing and provides support for DCL context in Caché.

22.6.1 Setup for Running TaskMan in a DCL Context in a Cache/VMS Environment

The following steps show you how to set up TaskMan to run in a DCL context in Cache/VMS (see Kernel patch XU*8.0*355).

NOTE: The following procedure is just an example and has to be modified for your site. You need to adjust the UIC [**100,20**] to match your system and indicate the location of the TaskMan directory.

1. Create TASKMAN user that runs the TaskMan jobs:

Figure 271: Create TASKMAN

```
ADD TASKMAN/OWNER=“SYSTEM MANAGER” -
/ACCOUNT=CACHE -
/PRIV=(NETMBX,TMPMBX) -
/DEFPRIV=(NETMBX,TMPMBX) -
/DEVICE=USER$/DIR=[TASKMAN]/LGICMD=LOGIN.COM -
/FLAGS=(DisCtlY,DisWelcome,DisReport,DisForce_Pwd_Change,DisPwdDic,DisPwdHis)
-
/PASS=TASK$MAN/UIC=[100,20]
```

2. Create the TASKMAN directory:

Figure 272: Create the TASKMAN Directory

```
Define/SYSTEM DHCP$TASKMAN USER$:[TASKMAN]
```

3. Create the system logical name for the directory with the COM files.

   NOTE: Be sure to also add to the **STARTUP$LOGICALS.COM** file.

   Figure 273: Create System Logical Name for the Directory with the COM Files

   ```
   Define/SYSTEM DHCP$TASKMAN USER$:[TASKMAN]
   ```

4. Create the queues, as explained in this manual.

   NOTE: Be sure to also add to the **STARTUP$DEFINE_QUEUES.COM** file.

   TaskMan submits jobs to the queue **TM$<*node*>**. Because you use "**run loginout**" to detach the execution, you do *not* need a large **JOB** limit here.

   Figure 274: Create System Logical Name for the Directory with the COM Files

   ```
   INIT/QUEUE/BATCH/OWNER=[TASKMAN] -
   /prot=(S:M,O:D,G:R,W:S)/JOB=5/AUTOSTART_ON=isfva2:: TM$isfva2
   ```

5. Load the following DCL command files into the [TASKMAN] directory:

   - **GET_METRIC.COM**
   - **LOGIN.COM**
   - **METRIC_SCHEDULE.COM**
   - **ZTM2WDCL.COM**
   - **ZTMS2WDCL.COM**

   These command files are located in the **cache-taskman** sub-directory in the Anonymous FTP site.

   NOTE: Get the files in ASCII mode.

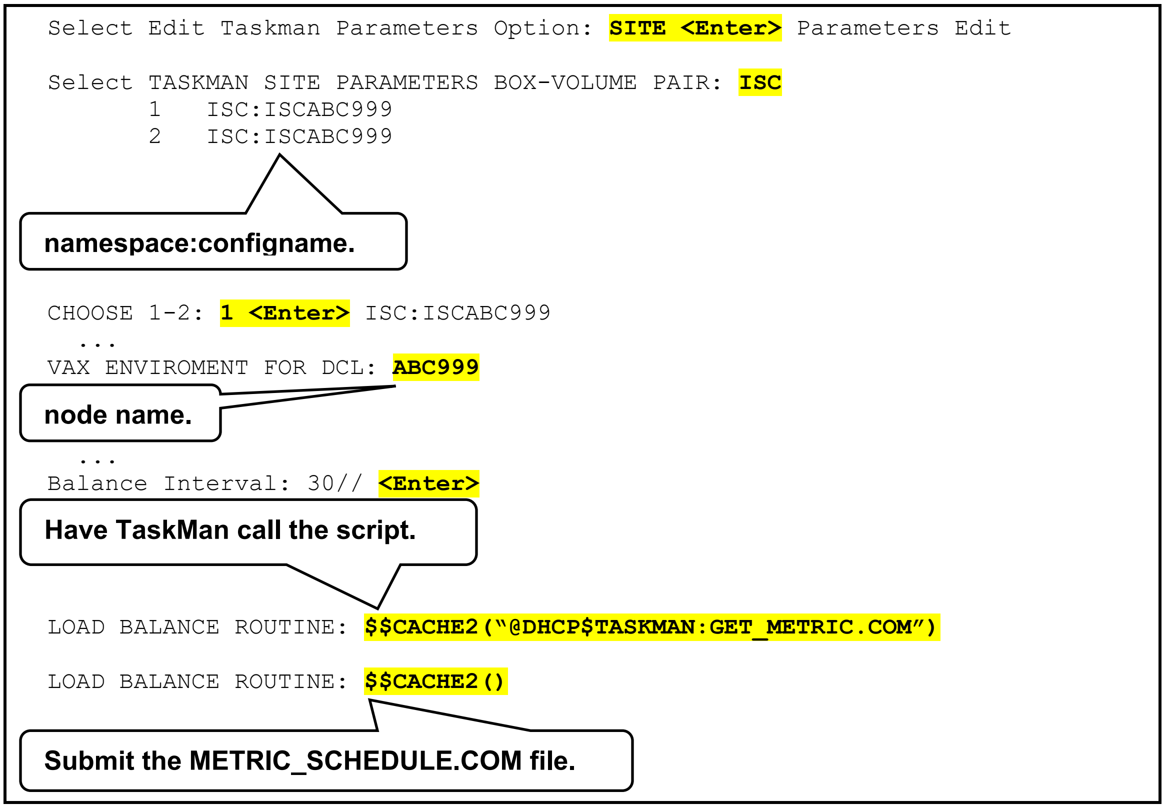
Figure 275: Sample User Dialog to Retrieve DCL Command Files

```
ABC999$SET DEF USER$:[TASKMAN]
ABC999$FTP <REDACTED>.VA.GOV
220 ABC999.<REDACTED>.VA.GOV FTP Server (Version 5.3) Ready.
Connected to FTP.<REDACTED>.VA.GOV.

Name (FTP.<REDACTED>.VA.GOV:fort): ANONYMOUS
331 Guest login OK, send ident as password.
Password: XXXXXXXXXX
230 Guest login OK, access restrictions apply.
FTP> CD CACHE-TASKMAN
FTP> LS
150 Opening data connection for USR$:[ANONYMOUS.CACHE-TASKMAN]*.*;*

GET_METRIC.COM
LOGIN.COM
METRIC_SCHEDULE.COM
ZTM2WDCL.COM
ZTMS2WDCL.COM

FTP> ASCII
200 TYPE set to ASCII.
FTP> GET GET_METRIC.COM
FTP> GET LOGIN.COM
FTP> GET METRIC_SCHEDULE.COM
FTP> GET ZTM2WDCL.COM
FTP> GET ZTMS2WDCL.COM
FTP> BYE
221 Goodbye.
```

NOTE: Repeat for each node in the TASKMAN SITE PARAMETERS (#14.7) file.

6. Edit TaskMan Parameters:

Figure 276: Sample User Dialog to Edit TaskMan Parameters



```
Select Edit Taskman Parameters Option: SITE <Enter> Parameters Edit

Select TASKMAN SITE PARAMETERS BOX-VOLUME PAIR: ISC
       1   ISC:ISCABC999
       2   ISC:ISCABC999

CHOOSE 1-2: 1 <Enter> ISC:ISCABC999
  ...
VAX ENVIROMENT FOR DCL: ABC999

  ...
Balance Interval: 30// <Enter>

LOAD BALANCE ROUTINE: $$CACHE2("@DHCP$TASKMAN:GET_METRIC.COM")

LOAD BALANCE ROUTINE: $$CACHE2()
```

namespace:configname.

node name.

Have TaskMan call the script.

Submit the METRIC_SCHEDULE.COM file.

22.6.2 How to Restart TaskMan when Running in a DCL Context

To manually restart TaskMan when TaskMan is running in a DCL context, you can either:

- Sign in as OpenVMS user TASKMAN and **DO RESTART^ZTMB**.
- Sign in from an OpenVMS account that has the **OPER** and **SYSPRV** privileges and **DO RESTART^ZTMB**. This submits the manager to run under the username TASKMAN.

In either case, however, do *not* use the **Restart Task Manager** [XUTM RESTART] option in the Kernel menus; it is *not* compatible with TaskMan in a DCL context.

Figure 277: ZTM2WDCL.COM Command File

```
$!-----------------------------------------------------------------------
$! ZTM2WDCL.COM - Cache Run Taskman in a DCL Context
$! * KERNEL 8 *
$!
$!  P1 is the Cache config that taskman should start in.
$!  P2 is the namespace that taskman should start in.
$!  P3 = null to START and 1 to RESTART
$!
$! This file is submitted to the queue to run and it
$!  builds and runs the TMP_pid.* files
$!
$! Build the file to run, can’t pass arguments with RUN
$ pid = F$GETJPI(“”,”PID”)
$ infile=“TMP_” + pid + “.ZTM”
$ outfile = “TMP_” + pid + “.log”
$ SAY = “write output”
$!
$ entry=“START”
$ if p3 .eq. 1 then entry=“RESTART”
$!
$! open and build the input file
$ OPEN/write output ‘infile’
$ SAY “$! Taskman temp file to run the Manager”
$ SAY “$! Delete this file if it is not open.”
$ SAY “$ set verify”
$ SAY “$ csession ““‘‘p1’”“ ““-U”“ ““‘‘p2’”“ ““‘‘entry’^%ZTM0”““
$ SAY “$ exit”
$ Close output
$!
$! If a log file is needed change _NLA0: to ‘outfile
$ name = “ZTMS_” + pid
$  run sys$system:loginout.exe -
       /input=‘infile -
       /output=_NLA0: -
       /detach /process=‘name
$!
$!      Wait for loginout to run it then delete the file.
$ wait 00:01
$!
$ del TMP_*.ZTM;1
$ exit
```

Figure 278: ZTMS2WDCL.COM Command File

```
$!----------------------------------------------------------------------
$! ZTMS2WDCL.COM - Cache Start Submanager with a DCL Context
$! * KERNEL 8 *
$! p1 is the Cache config name
$! p2 is the namespace to start.
$! p3 is NOT used. (VOL for DSM)
$!
$! This file is submitted to the queue to run and it
$!  builds and runs the TMP_pid file
$!
$! Build the file to run, can’t pass arguments with RUN
$ pid = F$GETJPI(“”,”PID”)
$ infile = “TMP_” + pid + “.ZTMS”
$ outfile = “TMP_” + pid + “.log”
$ SAY = “write output”
$!
$! open and build the input file
$ OPEN/write output ‘infile’
$ SAY “$! Taskman temp file to run a submanager”
$ SAY “$! Delete this file if it is not open.”
$ SAY “$ set verify”
$ SAY “$! ‘‘P1’ and ‘‘P2’”
$ SAY “$ csession ““‘‘p1’”“ ““-U”“ ““‘‘p2’”“ ““START^%ZTMS”““
$ SAY “$ exit”
$ Close output
$!
$! If a log file is needed change _NLA0: to ‘outfile
$ name = “ZTMS_” + pid
$  run sys$system:loginout.exe -
       /input=‘infile -
       /output=_NLA0: -
       /detach /process=‘name
$!
$!      Wait for loginout to run it then delete the file.
$ wait 00:01
$!
$ del TMP_*.ZTMS;1
$ exit
```

Figure 279: Example of OpenVMS User TASKMAN on ALPHA AXP Systems

```
Username: TASKMAN                          Owner:
Account:                                   UIC:    [50,20] ([DEV,TASKMAN])
CLI:      DCL                              Tables: DCLTABLES
Default:  USER$:[TASKMAN]
LGICMD:   LOGIN
Flags:  DisCtlY Restricted DisWelcome DisReport
Primary days:   Mon Tue Wed Thu Fri
Secondary days:                     Sat Sun
No access restrictions
Expiration:            (none)    Pwdminimum:  6   Login Fails:     0
Pwdlifetime:        180 00:00    Pwdchange:  19-NOV-1992 14:12
Last Login: 20-NOV-1992 10:34 (interactive), 20-NOV-1992 10:44 (non-
interactive)
Maxjobs:         0  Fillm:       300  Bytlm:        64000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:       300  JTquota:       4096
Prclm:          14  DIOlm:       900  WSdef:         2048
Prio:            4  ASTlm:       600  WSquo:         4096
Queprio:         0  TQElm:        10  WSextent:     16384
CPU:        (none)  Enqlm:      4096  Pgflquo:     100000
Authorized Privileges:
  CMKRNL TMPMBX OPER NETMBX
Default Privileges:
  CMKRNL TMPMBX OPER NETMBX
```

Figure 280: Example of OpenVMS TASKMAN Queue

```
ABC999$ SH QUE/FULL TM$ABC999

 Batch queue TM$ABC999, available, on ABC999:
   /BASE_PRIORITY=4 /JOB_LIMIT=50 /OWNER=[DEV,TASKMAN]
 /PROTECTION=(S:E,O:D,G:R,W:W)

ABC999$
```

23 TaskMan: System Management—Operation

This section describes how to operate TaskMan. It also discusses the following:

- TaskMan Management Menu
- Taskman Management Utilities
- Scheduling Options
- Taskman Error Log Menu
- Troubleshooting

23.1 TaskMan Management Menu

The **Taskman Management** [XUTM MGR] menu is the main point of entry into the TaskMan options. It contains the following options:

- **Schedule/Unschedule Options** [XUTM SCHEDULE]
- **One-time Option Queue** [XU OPTION QUEUE]
- **Taskman Management Utilities** [XUTM UTIL]
- **List Tasks** [XUTM INQ]
- **Dequeue Tasks** [XUTM DQ]
- **Requeue Tasks** [XUTM REQ]
- **Delete Tasks** [XUTM DEL]
- **Print Options that are Scheduled to run** [XUTM BACKGROUND PRINT]
- **Cleanup Task List** [XUTM TL CLEAN]
- **Print Options Recommended for Queueing** [XUTM BACKGROUND RECOMMENDED]

The **Taskman Management Utilities** [XUTM UTIL] submenu and the scheduling-related options are discussed later in this section. The options for listing, dequeuing, requeuing, deleting, and cleaning up tasks are discussed first.

23.1.1 List Tasks Option

Figure 281: List Tasks Option

```
SYSTEMS MANAGER MENU ...                                                  [EVE]
Taskman Management ...                                               [XUTM MGR]
   List Tasks                                                        [XUTM INQ]
```

Beginning with Kernel 8.0, the TASKS (#14.4) file (in **^%ZTSK**) is VA FileMan compatible (i.e., you can use VA FileMan to print out information about a task). However, the **List Tasks** [XUTM INQ] option also provides a way to examine tasks in the TASKS (#14.4) file. The **List Tasks** [XUTM INQ] option allows you to choose between several useful ways of selecting tasks. When you choose this menu, it presents you with the options shown in Figure 282:

Figure 282: List Tasks Option Submenu Options

```
                         List Tasks Option

                              All your tasks.
                              Your future tasks.
                              Every task.
                              List of tasks.
                              Unsuccessful tasks.
                              Future tasks.
                              Tasks waiting for a device.
                              Running tasks.

                         Select Type Of Listing:
```

Several choices only appear on the list when there are tasks in those collections to be displayed. Remember, the TASKS (#14.4) file can be Volume Set/CPU-specific. This means that the option can only display tasks from the TASKS (#14.4) file on the current Volume Set/CPU.

Holders of the ZTMQ security key see a slightly different list of selections. Instead of “All your tasks” and “Your future tasks” they see “All of one user’s tasks” and “One user’s future tasks.” These two selections are generic versions of those available to normal users. They allow the holder to see any user’s tasks and start by prompting the holder for the user whose tasks should be shown. Other than that, they are identical to the selections used by normal users.

Although each submenu option choice shows a different set of tasks, the format for the output is the same. Figure 283 is a sample display from the All your tasks suboption:

Figure 283: All your tasks Suboption—Sample of TaskMan Tasks Running

```
All tasks that you created...

2572: ALIVE^XINDEX, XINDEX of 1 routine.  Device QMS-17P.  VAH,KXX.
      From TODAY at 10:55,  By you.  Scheduled for TODAY at 12:05

End of listing.  Press RETURN to continue:
```

In the upper left-hand corner of each entry is the task number. What follows the task number is either an option name (e.g., XUTM QCLEAN) or a routine entry point (e.g., ERROR^ZTMZT) depending on whether the task was a queued routine or a queued option. This is generally followed by a description of the task. The device to which the task was queued (if any), along with the account in which the task was/is scheduled to run, complete the first line. The next line contains the time the task was created followed by an identification of the creator. In the case of tasks that requeue themselves, this date and time represents when the task was last requeued.

When the creator's **DUZ** number is *not* listed in the NEW PERSON (#200) file, the phrase "USER #" followed by the **DUZ** is substituted. Finally, the status of the task is shown.

REF: For a list and description of the status messages, see the "Troubleshooting" section.

Each of these submenu options are described in the topics that follow.

23.1.1.1 All your tasks Option

The **All your tasks** action (see Figure 283) displays every task in the TASKS (#14.4) file on the current Volume Set/CPU that you created. If you have no tasks scheduled, the option gives you the message "You have no tasks in this Volume Set's TASKS file."

23.1.1.2 Your future tasks Option

The **Your future tasks** action displays those tasks you created that are currently scheduled to run. If there are none, the option tells you.

"Every task" lists every task in the TASKS (#14.4) file.

23.1.1.3 List of tasks Option

The **List of tasks** action allows you to list one or more tasks by task number. You can specify individual tasks separated by commas along with ranges of tasks using a hyphen.

23.1.1.4 Unsuccessful tasks Option

The **Unsuccessful tasks** action lists three kinds of tasks:

- Rejected by the manager's validation process.
- Encountered an error while they were running.
- Unscheduled through the Dequeue Tasks option.

23.1.1.5 Future tasks Option

The **Future tasks** action shows all tasks that are in the Schedule List or the Waiting List. It does *not* show the tasks that are in the Job List. In other words, it shows all tasks that are scheduled to run but *not* those that are currently being run or those that are ready to be run. "Future Tasks" is *not* offered by the List Tasks option if the Schedule List and Waiting List are empty (an unlikely occurrence at most sites).

23.1.1.6 Tasks waiting for a device Option

The **Tasks waiting for a device** action shows just the Waiting List, which can be a useful way of isolating problem printers. If there are no tasks currently waiting for output devices to become available, the List Tasks option does *not* show this choice.

23.1.1.7 Running tasks Option

The **Running tasks** action shows tasks that are currently running.

REF: For a discussion of how TaskMan knows a task is running, see the "Troubleshooting" section.

23.1.2 Dequeue Tasks Option

Figure 284: Dequeue Tasks Option

```
SYSTEMS MANAGER MENU ...                                            [EVE]
Taskman Management ...                                          [XUTM MGR]
   Dequeue Tasks                                                 [XUTM DQ]
```

The **Dequeue Tasks** [XUTM DQ] option allows you to unschedule a task so that the task still exists in the TASKS (#14.4) file but is no longer in the Schedule, Waiting, or Job List. The process of unscheduling a task is called "dequeuing". This option allows you to dequeue any one task or range of tasks. A task that you dequeue has a status of **NOT QUEUED** in a List Tasks display.

The option first prompts you for the task number. Entering one question mark (**?**) gets you a short explanatory message, but entering two question marks (**??**) puts you in the **List Tasks** [XUTM INQ] option to find the task you are interested in dequeuing. When you leave the **List Tasks** [XUTM INQ] option, you automatically return to the task number prompt.

If you enter the number of a nonexistent task, the **List Tasks** [XUTM INQ] option tells you and then prompts you for another task number. If you enter the number of a task that does exist, the option displays the task and asks you if you are sure. Answering **NO** returns you to the task number prompt, whereas a **YES** dequeues the task and then returns you to the task number prompt.

You can also enter a list of tasks to be dequeued. The list can include single tasks separated by commas and ranges of tasks consisting of two numbers separated by a hyphen. After you enter the list, you are asked if you want to know the actual number of tasks in the list. You are then asked if you want a display of the actual tasks that are about to be dequeued.

Only holders of the ZTMQ security key can dequeue any task. Others can only dequeue their own tasks as identified by their **DUZ**.

23.1.3 Requeue Tasks Option

Figure 285: Requeue Tasks Option

```
SYSTEMS MANAGER MENU ...                                            [EVE]
Taskman Management ...                                          [XUTM MGR]
   Requeue Tasks                                                [XUTM REQ]
```

A benefit of the **Dequeue Tasks** [XUTM DQ] option is that it is completely *non*-destructive. If you dequeue a task and subsequently change your mind, you can use the **Requeue Tasks** [XUTM REQ] option to requeue the task exactly the way that it was. You can also use this option to change some of the details of a task that is already queued.

As with the **Dequeue Tasks** [XUTM DQ] option, you are first prompted for a task number with the same help available. Here, you can only enter a single task, *not* a range. The task is then displayed, and you are asked for a new run time with the default being either the original or current run time (whichever applies). The next question is "Do you wish to requeue this task to a device?", with the default depending on whether the task originally requested an output device. If you answer **YES**, the option asks you to specify an output device using the original output device (if there was one) as a default. The option also allows you to adjust the task's priority.

The task is requeued according to your specifications. Requeuing involves completely dequeuing the task so that your task does *not* run twice, making the changes you requested, and placing the task back on the Schedule List. Notice that the task is *not* dequeued until after you specify the changes you want to make. If you want to modify a task that may start running soon, it is usually a good idea to dequeue it first.

The ZTMQ security key affects this option in two ways

- Users who do *not* hold the security key are limited to requeuing only their own tasks.
- Users are *not* prompted to change the priority.

23.1.4 Delete Tasks Option

Figure 286: Delete Tasks Option

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Taskman Management ...                                             [XUTM MGR]
   Delete Tasks                                                    [XUTM DEL]
```

The **Delete Tasks** [XUTM DEL] option has the same structure as the **Dequeue Tasks** [XUTM DQ] option. The only difference is that where dequeuing a task just removes it from the lists (unschedules it); the **Delete Tasks** option also deletes the task from the TASKS (#14.4) file. When you have deleted a task, there is no reference to that task anywhere in TaskMan's files.

Only holders of the ZTMQ security key can delete any task. Others can only delete their own tasks as identified by their **DUZ**.

23.1.5 Cleanup Task List Option

Figure 287: Cleanup Task List Option

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Taskman Management ...                                             [XUTM MGR]
   Cleanup Task List                                          [XUTM TL CLEAN]
```

You can use the **Cleanup Task List** [XUTM TL CLEAN] option to remove a task entry from a task list for a job that is no longer running. This might happen when a process is forcibly exited, but TaskMan still believes the task is running. You can use this option to tell TaskMan which tasks you forcibly exited. TaskMan then removes those tasks from its list of running tasks.

23.2 Taskman Management Utilities

A submenu on the **Taskman Management** [XUTM MGR] menu, called **TaskMan Management Utilities** [XUTM UTIL] menu, provides several options to set up, monitor, and modify the TaskMan environment.

The **TaskMan Management Utilities** [XUTM UTIL] menu contains the following options:

- **Monitor Taskman** [XUTM ZTMON]
- **Check Taskman's Environment** [XUTM CHECK ENV]
- **Edit Taskman Parameters** [XUTM PARAMETER EDIT]
- **Restart Task Manager** [XUTM RESTART]
- **Place Taskman in a WAIT State** [XUTM WAIT]
- **Remove Taskman from WAIT State** [XUTM RUN]
- **Stop Task Manager** [XUTM STOP]
- **Taskman Error Log** [XUTM ERROR]
- **Clean Task File** [XUTM CLEAN]
- **SYNC flag file control** [XUTM SYNC]

These options are discussed in the sections that follow.

23.2.1 Monitor Taskman Option

Figure 288: Monitor Taskman Option

```
SYSTEMS MANAGER MENU ...                                          [EVE]
Taskman Management ...                                         [XUTM MGR]
   Taskman Management Utilities ...                           [XUTM UTIL]
      Monitor Taskman                                        [XUTM ZTMON]
```

The **Monitor Taskman** [XUTM ZTMON] option gives you a screen of information about the current state of TaskMan and offers you several ways to get more information. The monitor focuses on the current state of the manager itself and on the contents of the *non*-VA FileMan-compatible SCHEDULE file.

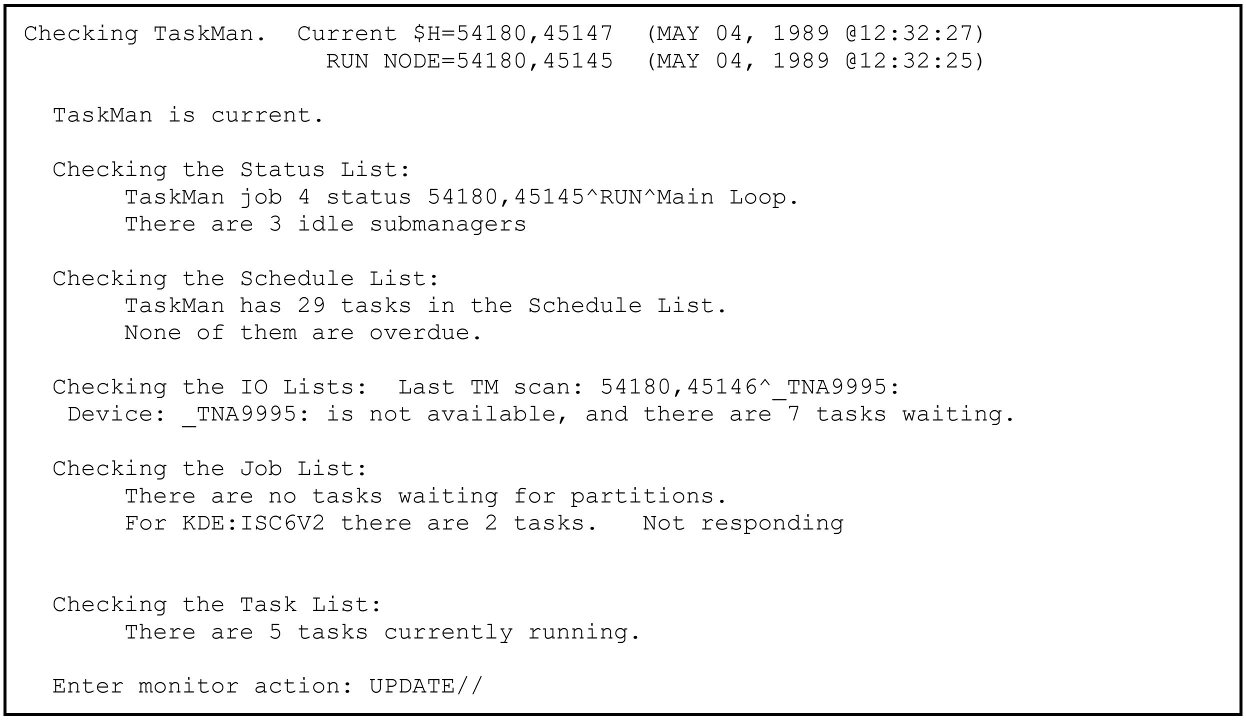
REF: for more information on the *non*-VA FileMan-compatible SCHEDULE file, see Section 23.5.1, "SCHEDULE File."

As you use this option, you acquire an intuitive understanding of how these lists should look and behave when your system is healthy. Spending the time using this option to get that intuition saves you troubleshooting time by helping you to notice problems sooner.

23.2.1.1 RUN Node

The first section of the "Monitor TaskMan" screen reports whether the manager is currently running on your machine, and if so, whether or not it is being delayed. This is accomplished by comparing TaskMan's **RUN** node to the M **$HOROLOG** variable. Under normal circumstances they should be within **15** seconds of each other, though certain conditions can cause a difference of up to **two** minutes. Any difference greater than that, however, is a sign that the manager is being delayed, typically by a problematic device or a recurring error. Of course, the manager is also likely to fall behind if the system is saturated to the point where all of the jobs on the system are slow. The last line of the first section evaluates the difference and guesses at the manager's current condition. The **$HOROLOG** values are translated into an external format for your convenience in understanding the values.

Figure 289: Sample Monitor TaskMan Screen



```
Checking TaskMan.  Current $H=54180,45147  (MAY 04, 1989 @12:32:27)
                     RUN NODE=54180,45145  (MAY 04, 1989 @12:32:25)

  TaskMan is current.

  Checking the Status List:
       TaskMan job 4 status 54180,45145^RUN^Main Loop.
       There are 3 idle submanagers

  Checking the Schedule List:
       TaskMan has 29 tasks in the Schedule List.
       None of them are overdue.

  Checking the IO Lists:  Last TM scan: 54180,45146^_TNA9995:
   Device: _TNA9995: is not available, and there are 7 tasks waiting.

  Checking the Job List:
       There are no tasks waiting for partitions.
       For KDE:ISC6V2 there are 2 tasks.   Not responding


  Checking the Task List:
       There are 5 tasks currently running.

  Enter monitor action: UPDATE//
```

23.2.1.2 Status List

The Status List is where each manager periodically reports its current status. The job number of the manager is reported both for ease of location on a system status report and also to distinguish between multiple managers (if there are more than one). Under normal circumstances, the manager removes its entry from the Status List when it shuts down, but if a manager stops abnormally (e.g., **RJD** or **FORCEX**) its entry is usually left on the list. The list is updated and cleaned out whenever a new manager is started or restarted.

The status of a manager consists of three parts:

- Date and time—This date and time should equal the **RUN** node's date and time, and like that node, it should be close to the current **$HOROLOG**.
- Manager's state.
- Description of special circumstances.

The manager can be in one of five states at any given time:

- **BALANCE**
- **ERROR**
- **PAUSE**
- **RUN**
- **WAIT**

RUN is the normal state, with a description of "Main Loop."

The manager's status is the most important piece of information the monitor gives, and it should always be the first thing checked when troubleshooting problems.

REF: For a detailed list and description of the possible state messages, see the "Troubleshooting" section.

23.2.1.3 Schedule List

The Schedule List always shows the number of tasks currently scheduled to run and checks the times for which they are scheduled to determine whether any of them should already have started. When many tasks are queued to run at the same time, it is *not* unusual for the manager to be a little late in sending off the last few.

When most of the tasks on the Schedule List are overdue, however, the manager is probably having problems keeping up. This is *not* a normal condition. If the problem is *not* a recurring error or a difficult output device, the most likely culprit is your default setup in the TASKMAN SITE PARAMETERS (#14.7) file. Another possible problem is that TaskMan is trapping many errors or trying to access a very slow link between Volume Sets. If the problem is error trapping, the Status List should regularly show the manager in an **ERROR** state. Also, remember that if the machine is saturated, all of the jobs on the system, including the manager, run slowly.

23.2.1.4 IO List

The **IO** List first shows the last time (**$H**) a submanager checked the list and the last device checked. The check generally shows how many tasks are waiting for each device in the **IO** List. The occasional remark "Allocated" means that a submanager has already noticed that the device is available and has allocated the device to a task using the Device Allocation List. Devices should only be allocated for a short time before the submanager opens the device, making it unavailable.

Understanding how the **IO** List works can make this particular check very useful. Submanagers handle the Device **IO** Lists. Unusual behavior in these lists usually points to device or submanager problems.

There are three fundamental things to look for with this check:

- When a device becomes available—The submanagers should notice and start a task running on that device. If the submanagers do *not* do this, it is probably time to start looking for problems with the submanagers.
- When a device is allocated—A submanager should quickly make it unavailable. If this fails to occur, the submanagers may be having problems. There can be extenuating circumstances (e.g., the system being very slow) that explain these occurrences.
- When many tasks are backed up waiting for the same device—Sometimes it is just because that device is busy. However, sometimes the device is off-line or out of paper.

23.2.1.5 Job List

The Job List is where tasks wait for partitions, so if many tasks are backed up here you know the submanagers are *not* picking them up. This can be caused by any of the following:

- A slow system.
- TaskMan reaching its job limit.
- TaskMan assigning tasks a priority that is too low for them to run.

Systems that are too busy back up in the Job List *not* the Schedule List. The Compute Server Job List is checked here and lets you know about tasks waiting to run on other CPUs and if the submanagers are *not* starting.

23.2.1.6 Task List

The Task List is where TaskMan keeps track of the tasks it has started running. Entries are set into this list when the submanagers start their tasks and are cleared when the tasks quit or cause errors to be trapped. **KILL**ing a task by forcing its process to exit in the middle of execution (using such vendor-specific tools as **RJD**, **RESJOB**, **FORCEX**, **KILLJOB**, etc.) does *not* give the submanager a chance to clear the task from the Task List, so the Task List can become inaccurate. If you frequently **KILL** jobs but want to keep your Task List accurate, you need to manually remove the obsolete entries. The exit action of the **KILL** off a users' job option

[XURESJOB] helps you identify and remove from the list of running tasks those you have forcibly exited.

23.2.1.7 Monitor Action Prompt

After summarizing the status of the manager and the principal lists of the SCHEDULE file, the monitor offers you a choice of actions. They are displayed if you enter a single question mark (**?**) at the “Enter monitor action:” prompt:

Figure 290: TaskMan Monitor Actions

```
Enter <RET> to update the monitor screen.
Enter ^ to exit the monitor.
Enter E to inspect the TaskMan Error File.
Enter S to see a system status listing.
Enter ? to see this message.
Enter ?? to inspect the tasks in the monitor’s lists.
```

These actions (see Figure 290) attempt to bring together those utilities used most often in response to seeing a monitor screen. Updating is the most commonly used choice since you often want to watch how the lists change over time. The TASKMAN ERROR file needs to be easily accessible, not only in case the manager enters an **ERROR** state, but also if a task that should take a long time to run leaves the Job List but never shows up in the Task List. This usually means the task hit an error and quit, which can be confirmed or disproved by a quick glance at the TaskMan Error Log. The System Status Report can be used to verify that tasks, submanagers, and the manager are indeed running as the monitor suggests.

Some actions at the Monitor Action prompt are *not* accessible when monitoring TaskMan from the manager’s account (using the direct-mode utility **D ^ZTMON**).

23.2.1.8 Inspecting the Tasks in the Monitor’s Lists

If you are in a non-library account, you can directly inspect the contents of the various lists. Do this by entering two question marks (**??**) at the “Enter monitor action:” prompt. You get the list of choices shown in Figure 291:

Figure 291: Options for Inspecting Tasks in the TaskMan Monitor’s Lists

```
  Help For Monitor Taskman Option

       Schedule List.
       Waiting Lists.
       One Waiting List.
       Job List.
       Task List.
       Link Lists.

  Select Type Of Listing:
```

These listings use the same format as that of the **List Tasks** action and show you the contents of the lists at the time you look at them. The One Waiting List listing prompts you to select a device, and the help for that prompt lets you see those devices that have tasks waiting. Many of these lists change very quickly. Thus, it is *not* unusual to enter the help with the intention of seeing the task that was shown by the main screen to be in the Job List, only to be informed by the help software that the Job List is now empty. These kinds of experiences are simply part of troubleshooting TaskMan.

While these monitor actions are useful, there are still times when you *must* leave the monitor to follow up on information you saw there. For example, you may want to check the list of unsuccessful tasks or to list a specific task; both these actions require using the **List Tasks** action.

Taken as a whole, the checks that make up the monitor can save you a lot of time in trying to evaluate TaskMan’s status. The example shown in Figure 289 is of a healthy, and *not* very busy, manager. Monitors at sites usually show considerably more activity, especially in the Waiting Lists.

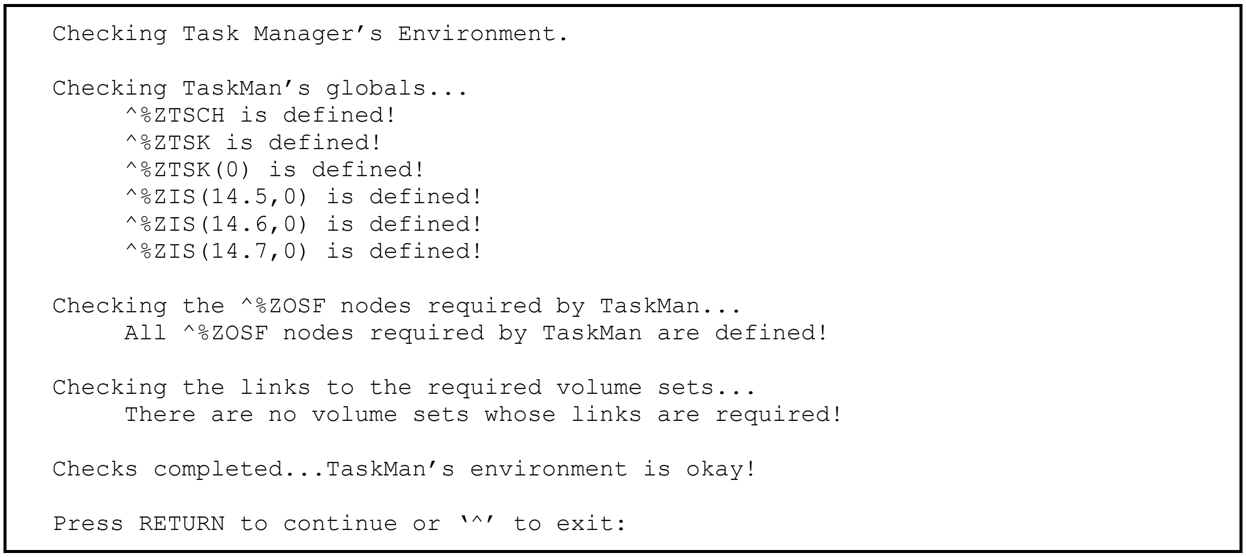
23.2.2 Check Taskman’s Environment Option

Figure 292: Check Taskman’s Environment Option

```
SYSTEMS MANAGER MENU ...                                                  [EVE]
Taskman Management ...                                               [XUTM MGR]
   Taskman Management Utilities ...                                 [XUTM UTIL]
      Check Taskman’s Environment                              [XUTM CHECK ENV]
```

The **Check Taskman’s Environment** [XUTM CHECK ENV] option presents two screens of information about TaskMan’s environment on the current CPU. The first screen (see Figure 293) performs all of the checks that the manager does whenever it starts, restarts, or encounters an error. The second screen (see Figure 294) shows what values the manager is using for its definition variables. This information can be very useful in pinpointing startup problems, in verifying that the manager is using the information you want it to use and in getting a general feel for how you have defined your system’s task management.

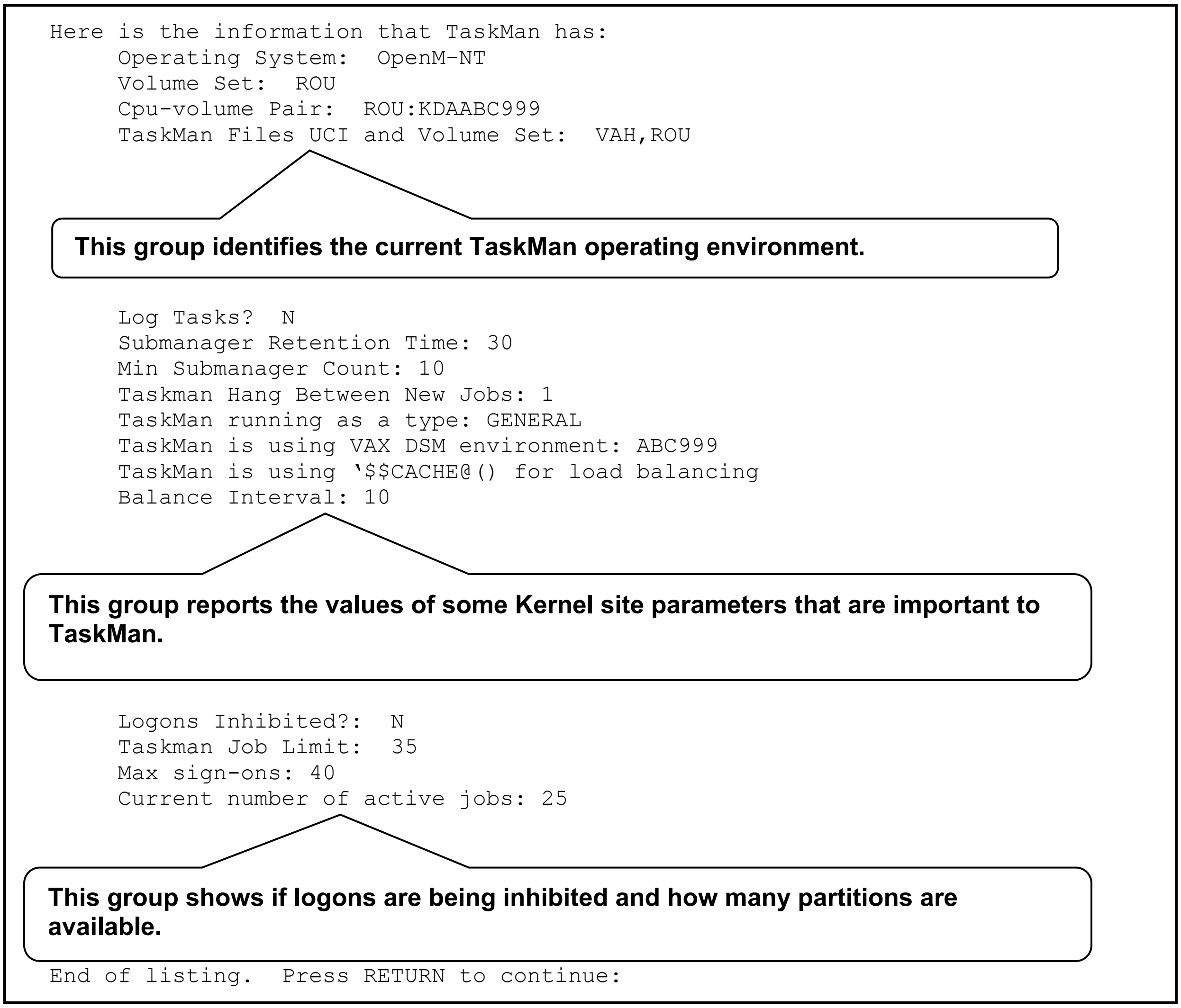
Figure 293: Check TaskMan’s Environment Option—First Screen



```
Checking Task Manager’s Environment.

Checking TaskMan’s globals...
     ^%ZTSCH is defined!
     ^%ZTSK is defined!
     ^%ZTSK(0) is defined!
     ^%ZIS(14.5,0) is defined!
     ^%ZIS(14.6,0) is defined!
     ^%ZIS(14.7,0) is defined!

Checking the ^%ZOSF nodes required by TaskMan...
     All ^%ZOSF nodes required by TaskMan are defined!

Checking the links to the required volume sets...
     There are no volume sets whose links are required!

Checks completed...TaskMan’s environment is okay!

Press RETURN to continue or ‘^’ to exit:
```

This first screen (see Figure 289) goes through each step that the manager goes through when it starts or restarts and reports the results. If your manager is failing to start, this screen should identify any problem with the environment.

Figure 294: Check TaskMan's Environment Option—Second Screen



```
Here is the information that TaskMan has:
     Operating System:  OpenM-NT
     Volume Set:  ROU
     Cpu-volume Pair:  ROU:KDAABC999
     TaskMan Files UCI and Volume Set:  VAH,ROU
```

This group identifies the current TaskMan operating environment.

```
     Log Tasks?  N
     Submanager Retention Time: 30
     Min Submanager Count: 10
     Taskman Hang Between New Jobs: 1
     TaskMan running as a type: GENERAL
     TaskMan is using VAX DSM environment: ABC999
     TaskMan is using ‘$$CACHE@() for load balancing
     Balance Interval: 10
```

This group reports the values of some Kernel site parameters that are important to TaskMan.

```
     Logons Inhibited?:  N
     Taskman Job Limit:  35
     Max sign-ons: 40
     Current number of active jobs: 25
```

This group shows if logons are being inhibited and how many partitions are available.

```
End of listing.  Press RETURN to continue:
```

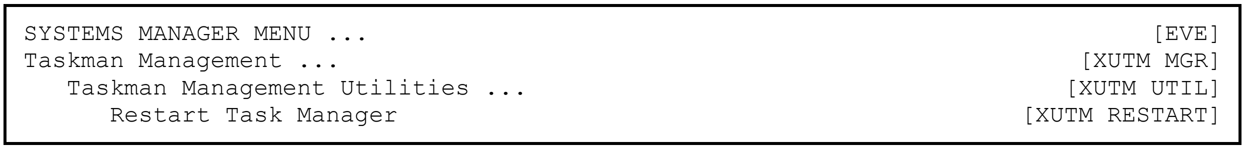
The second screen (see Figure 290) reports more information about the current TaskMan environment. The first group of four items identifies the current TaskMan operating environment. The next group of items reports the values of some Kernel site parameters that are important to TaskMan.

REF: These parameters, as well as all the other parameters that TaskMan uses, are described in detail in the "TASKMAN SITE PARAMETERS (#14.7) File" section in the "TaskMan: System Management—Configuration" section.

The last four items show if logons are being inhibited and how many partitions TaskMan currently has to work with. These values show how busy your system is, as well as how busy it can become. Their importance is also described in the discussion of parameters.

23.2.3 Restart Task Manager Option

Figure 295: Restart Task Manager Option

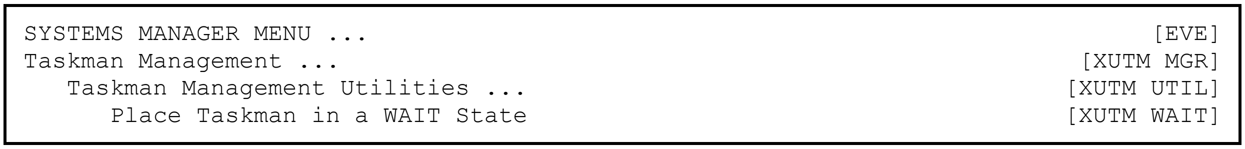


```
SYSTEMS MANAGER MENU ...                                        [EVE]
Taskman Management ...                                     [XUTM MGR]
   Taskman Management Utilities ...                       [XUTM UTIL]
      Restart Task Manager                             [XUTM RESTART]
```

The manager generally starts automatically when your system comes up. If the manager crashes or is stopped, you can use the **Restart Task Manager** [XUTM RESTART] option to restart it. The option first checks the **RUN** node and calculates whether it thinks the manager is currently running. If this option believes the manager is running, it asks you if you are sure you want to restart another TaskMan; you *must* answer **YES** to start the manager. If the **Restart Task Manager** [XUTM RESTART] option thinks the manager has stopped, it asks you for confirmation before jobbing out a new manager. If the **XUTM RESTART** option believes the manager to be active when you know for sure that it has failed, you can invoke the **Stop Task Manager** [XUTM STOP] option to prove to the **XUTM RESTART** option that the manager really has stopped. Then you are able to restart it.

23.2.4 Place Taskman in a WAIT State Option

Figure 296: Place Taskman in a WAIT State Option



```
SYSTEMS MANAGER MENU ...                                        [EVE]
Taskman Management ...                                     [XUTM MGR]
   Taskman Management Utilities ...                       [XUTM UTIL]
      Place Taskman in a WAIT State                       [XUTM WAIT]
```

The **WAIT** state (as described in the "Troubleshooting" section) is a condition in which the manager does nothing but wait for you to release it. Putting a stop to the manager's activities without actually shutting down the manager can often be very useful. For example, with the manager in a **WAIT** state, you can look at the tasks after they are queued but before the manager has a chance to validate them. This can help you isolate problems caused by the queuing process from those caused by the validation process. Another time you may want to create a **WAIT** state is before restarting a manager that has stopped. This prevents the manager from processing any tasks when it first starts up; the manager checks out its environment and then waits for your command to continue. The **Place Taskman in a WAIT State** [XUTM WAIT] option gives you a way to switch the manager's activities on and off without having to completely shut down and restart the manager.

When you select the **XUTM WAIT** option, you are also prompted with the question “Should active submanagers shut down after finishing their current tasks?”:

- If you answer **YES**, the submanagers on the current Volume Set/CPU quits when they finish a task instead of recycling.
- If you answer **NO**, the manager enters a **WAIT** state and the submanagers continue with their business.

If you also want to keep the submanagers from searching the Waiting List and the Job List for tasks, you need to explicitly say so at this prompt. This inhibition of the submanagers’ recycling remains in effect either until you remove the **WAIT** state or until a new manager starts or restarts, whichever comes first.

23.2.5 Remove Taskman from WAIT State Option

Figure 297: Remove Taskman from WAIT State Option

```
SYSTEMS MANAGER MENU ...                                              [EVE]
Taskman Management ...                                           [XUTM MGR]
   Taskman Management Utilities ...                             [XUTM UTIL]
      Remove Taskman from WAIT State                             [XUTM RUN]
```

The **Remove Taskman from WAIT State** [XUTM RUN] option simply undoes the effects of the **Place Taskman in a WAIT State** [XUTM WAIT] option, allowing the manager to process tasks and allowing the submanagers to recycle (if recycling had been inhibited).

23.2.6 Stop Task Manager Option

Figure 298: Stop Task Manager Option

```
SYSTEMS MANAGER MENU ...                                              [EVE]
Taskman Management ...                                           [XUTM MGR]
   Taskman Management Utilities ...                             [XUTM UTIL]
      Stop Task Manager                                         [XUTM STOP]
```

The **Stop Task Manager** [XUTM STOP] option gives you a clean way to stop the manager from within the menu system. This option also asks if you want the submanagers to shut down when they finish what they are doing.

NOTE: The **WAIT** state takes precedence. While the manager is in a **WAIT** state, not even **XUTM STOP** option affects it until after you invoke the **Remove Taskman from WAIT State** [XUTM RUN] option to release it from the **WAIT** state; after it is released, it shuts down.

This option should always be used to shut down TaskMan, rather than simply **KILL**ing the TaskMan process, which can leave the TaskMan globals in an improper state and even lose tasks.

23.2.7 SYNC flag file control Option

Figure 299: SYNC flag file control Option

```
SYSTEMS MANAGER MENU ...                                          [EVE]
Taskman Management ...                                       [XUTM MGR]
   Taskman Management Utilities ...                         [XUTM UTIL]
      SYNC flag file control                                [XUTM SYNC]
```

With the **SYNC flag file control** [XUTM SYNC] option, for any SYNC FLAG (#87) field entry in the TASKS (#14.4) file you can remove it from the file and delete all waiting tasks with the same SYNC FLAG. You can also choose **START NEXT**, which resumes running the series of tasks associated with that SYNC FLAG. This is useful when one task in a series of tasks that is synchronized with SYNC FLAG fails.

23.2.8 Clean Task File Option

The TASKS (#14.4) file grows every time a new task is queued. While the SAC requires applications to delete their tasks' entries when they complete, it is possible that older applications may *not* do this. Other tasks abort with errors; still others are rejected. The result is that **^%ZTSK** is always growing. Options are available that clean up the **^%ZTSK** global.

Figure 300: Clean Task File Option

```
SYSTEMS MANAGER MENU ...                                          [EVE]
Taskman Management ...                                       [XUTM MGR]
   Taskman Management Utilities ...                         [XUTM UTIL]
      Clean Task File                                      [XUTM CLEAN]
```

In unusual circumstances, you may need to clean the **^%ZTSK** global manually. Kernel provides the **Queuable Task Log Clean Up** [XUTM QCLEAN] option to regularly clean up the TASKS (#14.4) file in the background.

Only rarely are you *not* able to rely on the queued cleanup to perform this function. However, when necessary, you can use the interactive **Clean Task File** [XUTM CLEAN] option. First, **XUTM CLEAN** asks you if you are sure you want to clean out the old entries from the TASKS (#14.4) file. If you respond that you are, the option asks you how far back you want to keep old entries. The default is to keep old entries going back a week and to delete the older ones. After you provide this value, the option queues a task to do the cleanup. **XUTM CLEAN** *cannot* be queued.

23.2.9 Queuable Task Log Clean Up Option

The **Queuable Task Log Clean Up** [XUTM QCLEAN] option, resides on the **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu. This option allows you to purge all of the entries for tasks that are no longer queued (for whatever reason) and to purge the TaskMan Error Log. It is very useful to be able to queue the cleanup to automatically run each night; XUTM QCLEAN has been distributed to provide this feature. XUTM QCLEAN should *not* be run interactively; indeed, it is *not* available from any of TaskMan's menus. To queue this option, use the **Schedule/Unschedule Options** [XUTM SCHEDULE] option to queue it to run.

The date the **Queuable Task Log Clean Up** [XUTM QCLEAN] option starts purging the TASKS (#14.4) file is controlled by the DAYS TO KEEP OLD TASKS (#8) field parameter in the VOLUME SET (#14.5) file. A value of **seven** days is *recommended*. **XUTM QCLEAN** does *not* need an output device; therefore, you can leave that field blank. Once set up, the task automatically runs periodically, cleaning out inactive task entries that are older than the time period specified in the DAYS TO KEEP OLD TASKS (#8) field. If you want to run this on all of your machines, create an entry in the OPTION SCHEDULING (#19.2) file for each machine on which you want to run it.

23.3 Scheduling Options

TaskMan lets you, the site manager, schedule options that run regularly as tasks. Menu Manager and TaskMan work together to give you this ability. All you have to do is tell TaskMan which option you want to queue and how you want to queue it.

23.3.1 Which Options to Queue

The first requirement for queuing regards the option type. Only the run, print, and action types of options can be queued. The second requirement is that the option (if a run or action type) *must not* involve user input! There is nothing to prevent you from queuing an option of the wrong type or from queuing one that prompts the user for input but doing so results in a failed task. You *must* be conscious of the nature of the task when you consider creating one that performs an option. If the option itself does *not* run in the background, then queuing it is pointless. Even options that themselves queue tasks probably *cannot* be queued, because most of these ask the user for an output device or a run time.

Software applications can make recommendations for scheduling of options. This is a great help to site managers.

REF: Recommendations for scheduling Kernel options can be found in the *Kernel Installation Guide* and the *Kernel 8.0 and Kernel Toolkit 7.3 Technical Manual*.

23.3.1.1 PARENT OF QUEUABLE OPTIONS Menu

Some options that are intended to be queued are *not* intended to be run interactively, so placing such options on a user menu could cause problems. The **Parent of Queuable Options** [ZTMQUEUABLE OPTIONS] menu, a menu-type option, has no parent in the menu tree and is intended to be used as the parent of all such options.

23.3.1.2 Printing Options Recommended to Run and Scheduled to Run

Figure 301: Print Options Recommended for Queueing and Print Options that are Scheduled to Run Options

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Taskman Management ...                                             [XUTM MGR]
  Print Options Recommended for Queueing         [XUTM BACKGROUND RECOMMENDED]
  Print Options that are Scheduled to run              [XUTM BACKGROUND PRINT]
```

23.3.1.2.1 Print Options Recommended for Queueing Option

The **Print Options Recommended for Queueing** [XUTM BACKGROUND RECOMMENDED] option displays all options in the OPTION SCHEDULING (#19.2) file that are *recommended* for scheduling by the option's developer.

23.3.1.2.2 Print Options that are Scheduled to run Option

The **Print Options that are Scheduled to run** [XUTM BACKGROUND PRINT] option lists all currently scheduled options on your system. By comparing these two reports, you can see if any options recommended for scheduling are *not* scheduled on your system (and vice-versa).

23.3.1.3 Schedule/Unschedule Options

Figure 302: Schedule/Unschedule Options Option

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Taskman Management ...                                             [XUTM MGR]
   Schedule/Unschedule Options                                [XUTM SCHEDULE]
```

The **Schedule/Unschedule Options** [XUTM SCHEDULE] option is a straightforward VA ScreenMan edit option, and it allows you to schedule and unschedule options. After you select the option to schedule, you are prompted for information about the task you want to set up. You can edit the following fields in the OPTION SCHEDULING (#19.2) file:

- QUEUED TO RUN AT WHAT TIME (#2) (see Section 23.3.1.4)
- DEVICE FOR QUEUED JOB OUTPUT (#3) (see Section 23.3.1.7)
- QUEUED TO RUN ON VOLUME SET (#5) (see Section 23.3.1.8)
- RESCHEDULE FREQUENCY (#6) (see Section 23.3.1.9)
- SPECIAL QUEUEING (#9) (see Section 23.3.1.11)
- TASK PARAMETERS (#15) (see Section 23.3.1.10)

The cross-references on these fields make calls to TaskMan's API to update the TASKS (#14.4) file and **^%ZTSCH**.

NOTE: In order to queue a task, its SCHEDULING RECOMMENDED (#209) field in the OPTION (#19) file *must* be set to **YES**.

23.3.1.4 Queued to Run At What Time

To queue an option, select the option and enter a time at least **two minutes** in the future into the QUEUED TO RUN AT WHAT TIME (#2) field in the OPTION SCHEDULING (#19.2) file. When you enter a time (and date) for the task to run, the task is immediately put on the Schedule List for that time.

23.3.1.5 How to Delete a Regularly Scheduled Task

Deleting a scheduled task is as simple as entering the at-sign (**@**) at the QUEUED TO RUN AT WHAT TIME (#2) field. TaskMan then searches the current TASKS (#14.4) file for the task that corresponds to the entry in the OPTION SCHEDULING (#19.2) file and deletes it.

If your system has multiple copies of the TaskMan globals, you *must* use the **Schedule/Unschedule Options** [XUTM SCHEDULE] option on the same Volume Set/CPU where your task originated, when you delete the task. Otherwise, the future task in the TASKS (#14.4) file is *not* found (and deleted) when you enter an at-sign (**@**) in the QUEUED TO RUN AT WHAT TIME (#2) field.

23.3.1.6 How to Requeue a Regularly Scheduled Task

Requeuing merely involves placing a new value in the QUEUED TO RUN AT WHAT TIME (#2) field. When you do this, the currently scheduled task is deleted (exactly as described above when deleting a scheduled task). Then, a new task is created at the new time to replace the previously scheduled task.

If your system has multiple copies of the TaskMan globals, you *must* use the **Schedule/Unschedule Options** [XUTM SCHEDULE] option on the same Volume Set/CPU where your task originated, when you requeue the a task. Otherwise, the existing future task in the TASKS (#14.4) file is *not* found (and deleted) when you enter a new time in the QUEUED TO RUN AT WHAT TIME (#2) field.

23.3.1.7 Device For Queued Job Output

The DEVICE FOR QUEUED JOB OUTPUT (#3) field in the OPTION SCHEDULING (#19.2) file is where you can give the task an output device. For print (Report) type options this is obviously mandatory; for run or action types you need to consider if the option needs an output device. Modifying this value for an already-scheduled task merely causes a direct change to the currently scheduled task:

- Tasks with an output device are assigned a process name of:

  Task ####

  Where #### is the task number.

- Tasks with no output device are assigned a process name of:

  BTask ####

  (with **B** meaning background)

23.3.1.8 Queued To Run On Volume Set

Use the QUEUED TO RUN ON VOLUME SET (#5) field in the OPTION SCHEDULING (#19.2) file to designate a Volume Set or CPU for the task other than your current one. This field is only useful for options that do *not* have a device selected because most devices are tied to a CPU, and thus, the task *must* run on the CPU that has that device.

Modifying this value for an already-scheduled task merely causes a direct change to the currently scheduled task.

Running a task on each CPU for a given option may at times be useful (e.g., the **Non-interactive Build Primary Menu Trees** [XQBUILDTREEQUE] option). In such cases, make multiple entries in the OPTION SCHEDULING (#19.2) file and use the QUEUED TO RUN ON VOLUME SET (#5) field to specify the Volume Set/CPU where each scheduled task should run.

If you leave the DEVICE FOR QUEUED JOB OUTPUT (#3) field blank, the task that performs the option runs without a device (or tries to). If you also leave the QUEUED TO RUN ON VOLUME SET (#5) field blank, the task runs on the current CPU without a device. If you fill in both fields, TaskMan uses the value of the QUEUED TO RUN ON VOLUME SET (#5) field, unless overridden by the VOLUME SET(CPU) (#1.9) field in the DEVICE (#3.5) file entry of the selected device.

23.3.1.9 Reschedule Frequency

Whenever a task starts running an option, it looks to see what is in the RESCHEDULE FREQUENCY (#6) field in the OPTION SCHEDULING (#19.2) file. If the field is blank, the option does *not* reschedule itself. If you have filled in this field, the task uses the value you placed in the field to figure out when you want it to run next. Then it updates the QUEUED TO RUN AT WHAT TIME (#2) field to reflect the new scheduled time. When this field is updated, the next task in the sequence is scheduled.

If you change the existing value in the RESCHEDULE FREQUENCY (#6) field, the new increment is used beginning after the next time the option runs.

There are several formats you can use in this field:

- Every "**n**" seconds.
- Hours.
- Days.
- Months (incremental).
- A particular day of the month.
- A list of times every "**n**" months.

REF: For a list of the code formats for the RESCHEDULE FREQUENCY (#6) field, see the "Special Queueing" section.

For the incremental scheduling frequencies (every n seconds, hours, days, or months), the increment is added to the scheduled date and time in the QUEUED TO RUN AT WHAT TIME (#2) field to determine when the task should run next. As of Kernel 8.0, if the incremented time is in the past, however, TaskMan keeps adding the increment until a future time is reached, only then does it reschedule the task.

23.3.1.10 Task Parameters

Use the TASK PARAMETERS (#15) field in the OPTION SCHEDULING (#19.2) file to pass data to a scheduled option. TASK PARAMETERS holds a string that is passed to scheduled jobs through the **ZTQPARAM** variable. Ideally, the developer of an option that uses the TASK PARAMETERS string should describe the format and meaning of the string in the option's DESCRIPTION (#3.5) field in the OPTION (#19) file.

23.3.1.11 Special Queueing

Use the SPECIAL QUEUEING (#9) field in the OPTION SCHEDULING (#19.2) file to designate which option is scheduled to be run by TaskMan.

NOTE: In order to queue a task, its SCHEDULING RECOMMENDED (#209) field in the OPTION (#19) file *must* be set to **YES**.

Valid values are listed in Table 45:

Table 45: Special Queueing Field Settings

Value	Option Description
S	**STARTUP—**TaskMan queues the job to run whenever the TaskMan/computer is started (i.e., at System Boot). If you want to the run the startup option on multiple CPUs, make multiple entries in the OPTION SCHEDULING (#19.2) file and use the QUEUED TO RUN ON VOLUME SET (#5) field to specify on what Volume Set/CPU each should run.
SP	**STARTUP/PERSISTENT—**TaskMan queues the job as it does for “STARTUP. It marks it as a “PERSISTENT” task to be restarted if it stops unexpectedly.
P	**PERSISTENT—**TaskMan runs it on its normal schedule, marking it as Persistent. TaskMan restarts the task if it stops unexpectedly. If the task completes in a normal fashion it is treated like any other regularly scheduled task and it is rescheduled based on the value in the RESCHEDULING FREQUENCY (#6) field in the OPTION SCHEDULING (#19.2) file.

Table 46: Option Scheduling Frequency Code Formats

<table>
<tr><th>Code</th><th>Frequency</th></tr>
<tr><td>nS</td><td>Every <i>n</i> seconds.</td></tr>
<tr><td>nH</td><td>Every <i>n</i> hours.</td></tr>
<tr><td>nD</td><td>Every <i>n</i> days.</td></tr>
<tr><td>nM</td><td>Every <i>n</i> months.</td></tr>
<tr><td>day[@time]</td><td>Day of week (for Day codes, see Table 47).</td></tr>
<tr><td>D[@time]</td><td>Every weekday.</td></tr>
<tr><td>E[@time]</td><td>Every weekend day (Sat,Sun).</td></tr>
<tr><td>nM(entry[,entry[,...]])</td><td>Every <i>n</i> months, at each entry in the parameter list; the entries in the parameter list (for every <i>n</i> months only) can be:
<table>
<tr><th>Entry Format</th><th>Frequency</th></tr>
<tr><td>dd[@time]</td><td>Day of month (e.g., 15).</td></tr>
<tr><td>nday[@time]</td><td>Nth day of week in month (e.g., 1W,3W).</td></tr>
<tr><td>L[@time]</td><td>Last day of month.</td></tr>
<tr><td>Lday[@time]</td><td>Last specific DAY in month,
(e.g., LM,LT,LW...).</td></tr>
</table>
</td></tr>
</table>

Table 47: Day Codes Used in Option Scheduling Frequency Code Formats

Day Code	Description
M	Monday
T	Tuesday
W	Wednesday
R	Thursday
F	Friday
S	Saturday
U	Sunday

Table 48: Examples of Option Scheduling Frequency Code Formats

Code	Frequency
12H	Every 12 hours.
14D	Every 14 days.
1M(1,15)	First and 15th of the month.
1M(L@23:45)	Last day of the month at 11:45 pm.
1M(LS)	The last Saturday of the month.
3M(15@12:00,L@12:00)	Noon (on the 15th and last days), every 3 months.
W@4pm	Each Wednesday at 4 pm.
D	Each weekday.

23.3.1.12 Problems with Scheduled Options

Once an option has been put on a schedule, it stays on that schedule unless one of the following happens:

- You delete the task.
- The running task aborts while setting up the next task in the sequence; the schedule sequence is broken.
- You dequeue the task that is scheduled to run the option. You *must* either requeue the task or use the **Schedule/Unschedule Options** [XUTM SCHEDULE] option to start the cycle over.
- You change the value in the RESCHEDULING FREQUENCY (#6) field in the OPTION SCHEDULING (#19.2) file. The new increment is used beginning after the next time the option runs.

- You change the value in the QUEUED TO RUN AT WHAT TIME (#2). The currently scheduled task is unscheduled and a new one is scheduled for the time you specify.

Another peculiarity in this process involves using a monthly scheduling frequency. What should happen if on January 31st you queue an option and give it a monthly scheduling frequency? Other months lack a 31st day. In this situation, the task pretends there is a 31st day in every month. To avoid this, you can use the RESCHEDULING FREQUENCY (#6) field in the OPTION SCHEDULING (#19.2) file code **1M(L@time)**.

23.3.1.13 One-time Option Queue Option

Figure 303: One-time Option Queue Option

```
SYSTEMS MANAGER MENU ...                                                [EVE]
Taskman Management ...                                              [XUTM MGR]
   One-time Option Queue                                     [XU OPTION QUEUE]
```

Use the **One-time Option Queue** [XU OPTION QUEUE] option to run at a special time one day without affecting its established schedule. It queues a task to run once, without affecting the option’s normal schedule in any way. This lets you handle the condition where you have an option queued to run periodically and you would like to queue it once to run at an irregular time without affecting its normal periodic schedule.

23.4 Taskman Error Log Menu

The manager and submanagers keep track of all errors caused by their own software or by the tasks they start. They log their own errors in two places:

- ERROR LOG (#3.075) file
- TaskMan Error Log

Those errors caused by tasks are also recorded in the entries of the tasks themselves and can be seen with any of the various task listing options (List Tasks, TaskMan User, etc.). Just as there are options to display and purge the ERROR LOG (#3.075) file, there are options to do the same for the TaskMan Error Log.

When the **Queuable Task Log Cleanup** [XUTM QCLEAN] option cleans tasks from the TASKS (#14.4) file, it also cleans any corresponding entries in the TaskMan Error Log since it is hard to make sense of an error log entry without the task data.

Kernel strongly recommends that you report new errors to your OIFOs and follow up to ensure expeditious patching. If you do this, over time the number of errors occurring on your system diminishes. This also improves the value of the various error logging systems as indicators of significant events deserving investigation.

Allocation and store errors are often *not* logged in Kernel’s ERROR LOG (#3.075) file because the process of logging errors is complicated and usually requires the use of local variables. Local variables take up space and there is no excess space when these errors occur. However, TaskMan makes its simple entries in the TaskMan Error Log prior to calling the Kernel error logging utility. Thus, these errors are often recorded in the TaskMan Error Log, but *not* Kernel’s. You are encouraged to carefully monitor both places.

23.4.1 Show Error Log Option

Figure 304: Show Error Log Option

```
SYSTEMS MANAGER MENU ...                                                       [EVE]
Taskman Management ...                                                    [XUTM MGR]
   Taskman Management Utilities ...                                      [XUTM UTIL]
      Taskman Error Log ...                                             [XUTM ERROR]
         Show Error Log                                            [XUTM ERROR SHOW]
```

The **Show Error Log** [XUTM ERROR SHOW] option displays the errors currently stored in the TaskMan Error Log, showing the date and time that the error occurred in a readable format and showing the error message. After the listing, the option gives the number of errors in the error log.

Errors stored in the TaskMan Error Log historically are also cross-referenced to the TASKS (#14.4) file, linking tasks to the errors they cause.

23.4.2 Clean Error Log Over Range Of Dates Option

Figure 305: Clean Error Log Over Range Of Dates Option

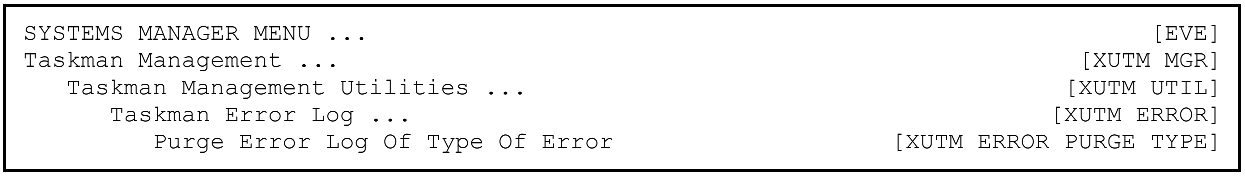
```
SYSTEMS MANAGER MENU ...                                                       [EVE]
Taskman Management ...                                                    [XUTM MGR]
   Taskman Management Utilities ...                                      [XUTM UTIL]
      Taskman Error Log ...                                             [XUTM ERROR]
         Clean Error Log Over Range Of Dates            [XUTM ERROR LOG CLEAN RANGE]
```

After prompting for a “First date to purge:” and a “Final date to purge:”, the **Clean Error Log Over Range Of Dates** [XUTM ERROR LOG CLEAN RANGE] option removes the entries for all errors that occurred on and between the two dates. It prints the number of entries removed. If the first date is *not* earlier than the final date, no entries are removed.

Use this option to delete all but recent errors that deserve your attention. It is better to resolve specific kinds of errors as you encounter them. However, if there is a period during which you *cannot* resolve them fast enough to keep the log clean, this option helps you focus on the recent ones.

23.4.3 Purge Error Log Of Type Of Error Option

Figure 306: Purge Error Log Of Type Of Error Option



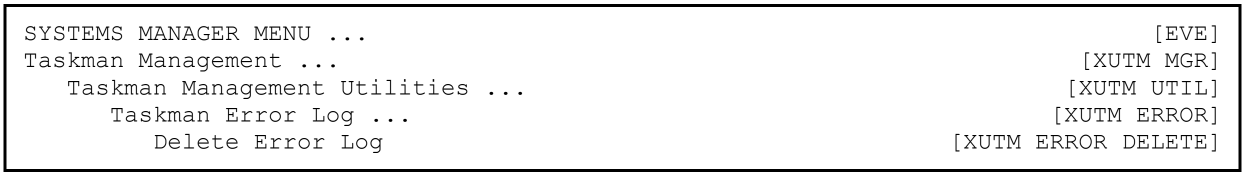
```
SYSTEMS MANAGER MENU ...                                                      [EVE]
Taskman Management ...                                                   [XUTM MGR]
   Taskman Management Utilities ...                                     [XUTM UTIL]
      Taskman Error Log ...                                            [XUTM ERROR]
         Purge Error Log Of Type Of Error                   [XUTM ERROR PURGE TYPE]
```

With the **Purge Error Log Of Type Of Error** [XUTM ERROR PURGE TYPE] option you can delete from the TaskMan Error Log all entries for an error of a specific type. In fact, this option uses the M contains operator (**[**); therefore, it removes every error whose message contains your input as a substring. For example, you can remove every error that occurred in a certain routine or even every error whose message contains a **Q**. After performing the purge, the option shows you how many entries were removed.

This option is the best way to keep the log clean. As you resolve certain kinds of errors and prevent them from happening again, you can remove all errors of that kind from the log. This leaves behind only those errors you have *not* resolved, helping you focus on the problems that remain.

23.4.4 Delete Error Log Option

Figure 307: Delete Error Log Option



```
SYSTEMS MANAGER MENU ...                                                      [EVE]
Taskman Management ...                                                   [XUTM MGR]
   Taskman Management Utilities ...                                     [XUTM UTIL]
      Taskman Error Log ...                                            [XUTM ERROR]
         Delete Error Log                                       [XUTM ERROR DELETE]
```

The **Delete Error Log** [XUTM ERROR DELETE] option completely deletes all errors in the TaskMan Error Log. If the error log is cleaned and purged as described above, you rarely need to use this option.

23.5 Troubleshooting

The information given in this section *cannot* be used by application developers in their code. It is provided to help site managers troubleshoot problems with tasks and TaskMan. Consider this section a reference to TaskMan's global structure and messages.

23.5.1 SCHEDULE File

The **^%ZTSCH** global holds the *non*-VA FileMan-compatible SCHEDULE file, which consists of independent lists and nodes (see Table 49). This is where TaskMan processes tasks. This structure is *not* supported for use by application software. All task manipulation *must* be done through approved options and entry points. These structures *must* be free to change from version to version to easily adapt and meet the changing needs of VistA. On the following pages is an example of a global that contains one of each type of node used by TaskMan:

The initial node was used to create **^%ZTSCH** before TaskMan was active, so that the global type and protection could be assigned.

Table 49: ^%ZTSCH (SCHEDULE File) Nodes

^%ZTSCH Node	Description
^%ZTSCH(next run time, task #)	This node stores the Schedule List. The task # corresponds to an entry in the TASKS (#14.4) file, and the next run time is computed from the value in the sixth ^-piece of the entry's **0** node (and is the total number of seconds contained in the next run time's **$H** translation). If the Schedule List entry equals a device name, the entry was *not* created through the Program Interface.
^%ZTSCH("C")	This node stores the Compute Server Job List (C list). This list holds tasks that are ready to be run by submanagers on specific Compute Servers. A submanager cross-Volume Set jobbed to a Compute Server only runs tasks under this list for the Compute Server on which it is running and does *not* process the Device Waiting List or the Job List. The Volume Set, next run time, task **#**, and device **$IO** are stored here.
^%ZTSCH("DEV")	This node stores the Device Allocation List. This list is used by TaskMan to coordinate its allocation of devices to tasks. The presence of a node indicates that TaskMan has already allocated this device to a specific task that has *not* yet gained ownership of it. It tells TaskMan *not* to give the device to another task. When the task for whom the allocation node was established gains ownership of the device or fails due to possession by some interactive job, the

^%ZTSCH Node	Description
	node is **KILL**ed off. The **$H** value is used in case the task fails to remove its own node for some reason; after two minutes TaskMan **KILL**s the node on its next idle loop.
^%ZTSCH("ER")	This node stores the TaskMan Error Log.
^%ZTSCH("ES")	This node stores the Error Screens.
^%ZTSCH("IDLE")	This node is used to ensure that the Manager's idle loop activities are spaced out correctly in case multiple managers are being run in the same environment.
^%ZTSCH("IO")	This node stores the Device Waiting List. The device **$IO** value is the value for the task's device and should *not* be the **$IO** of a spool or host file device. The run time subscript (the total number of seconds contained in the run time's **$H** translation) prioritizes the tasks that should have started the longest time ago. The submanagers use the top node to space out access to the list and the last device so that only one submanager at a time is checking the list, and so that checks that find all devices still busy are followed by a short waiting period before the list is checked again.
^%ZTSCH("JOB")	This node stores the Job List. This list holds tasks that are ready to be run by submanagers. The run time is the total number of seconds contained in the run time's **$H** translation, and the task # and device **$IO** are what you would expect.
^%ZTSCH("LINK")	This node stores the Link Lists. The **LINK** node itself is only present when a link is down. It is used to time the checks that occur every fifteen minutes. The second level nodes should always be present with the current information on each of the CPUs and Volume Sets.
^%ZTSCH("LOAD", load rating)	This node is used to balance the CPU load among the various managers that work out of the current TASKS and Schedule files. It identifies the CPU that most recently checked its rating and decided to run. Managers more loaded (a lower rating) than this one wait to allow this Manager to pick up more of its share of the load.
^%ZTSCH("LOADA")	This node stores the Load List. This list records the ratings for all the CPUs with managers processing this TASKS file. The first ^-piece,

^%ZTSCH Node	Description
	which flags the managers that decide to wait to balance the load, is used to tell the submanagers on those CPUs that they, too, should wait.
^%ZTSCH(“LOGRSRC”)	This node flags whether submanagers should log resources for the capacity management software. This node is set for every Volume Set whenever the LOG RESOURCE USAGE? field of the KERNEL SYSTEM PARAMETERS (#8989.3) file is edited. A cross-reference keeps the **^%ZTSCH(“LOGRSRC”)** node in synchronization with the LOG RESOURCE USAGE? field.
^%ZTSCH(“NO-OPTION”)	If set, this node stops the submanagers from running any scheduled options. This is for the KIDS install process.
^%ZTSCH(“RUN”)	This node is where the Manager periodically stamps the current time, leaving a way to determine whether it is currently active. Invoking the **Stop Task Manager** [XUTM STOP] option removes this node (see Figure 308).
^%ZTSCH(“STARTUP”, UCI, option #)	This node holds the Startup List. This list holds the internal number of all options that are specially queued to run every time the Manager starts up. The **$HOROLOG** value reflects when the option was placed on this list.
^%ZTSCH(“STATUS”, $J of Manager)	This node holds the Status List. This list holds the periodically updated entries for each Manager active on your machine and reflects each Manager’s own perception of its current state.
^%ZTSCH(“STOP”)	This node prevents submanagers from running. While it is present, managers do *not* start new submanagers, submanagers waiting for tasks quit immediately, and those currently running tasks quit as soon as the tasks finish.
^%ZTSCH(“SUB”)	This node counts the number of submanagers waiting for new tasks. It is updated regularly by submanagers as they run tasks. The Manager uses this value to decide whether to **JOB** out new submanagers and adjusts its value during the idle loop whenever it believes it to be inaccurate.
^%ZTSCH(“TASK”, task #)	This node holds the tasks TaskMan believes are currently running. Since entries are cleaned up when tasks quit or encounter errors, those that are forcibly exited by the system manager are left

^%ZTSCH Node	Description
	on the list even though they are *not* running. The Manager clears the list whenever the system starts up, and the system manager can manually remove inaccurate entries by using the exit action of the **KILL off a users' job** [XURESJOB] option. The task data stored at each node allows TaskMan to list the tasks even when they clean out their TASKS (#14.4) file records when they start instead of when they quit.
^%ZTSCH("UPDATE", $J of Manager)	This node, records when the Manager last updated its local information about the site parameters. This node is **KILL**ed whenever the Manager should update (e.g., site parameters are changed).
^%ZTSCH("WAIT")	This node puts the Manager into a **WAIT** state.

Figure 308: ^%ZTSCH Global Structure

```
^%ZTSCH= ““
^%ZTSCH(next run time, task #)= “”
^%ZTSCH(next run time, task #)= (D1) device IOP value
^%ZTSCH(“C”, volume set)= count
^%ZTSCH(“C”, volume set, next run time, task #) = device $IO
^%ZTSCH(“DEV”, device $IO)= $H when device was allocated for a specific
                                         ==>task
^%ZTSCH(“ER”)= “A1” or “”
^%ZTSCH(“ER”, $H when error happened)= error message
^%ZTSCH(“ER”, $H when error happened, 0)= context of error
^%ZTSCH(“ES”, error screen, 0)= “”
^%ZTSCH(“ES”, error screen, 1)= screened errors count
^%ZTSCH(“IDLE”)= $H when the Manager’s idle loop checks were last performed
^%ZTSCH(“IO”)= $H when device waiting list was last checked without finding
             ==> an available device^ $IO of last device tried
^%ZTSCH(“IO”, device $IO)=device type
^%ZTSCH(“IO”, device $IO, run time, task #)= “”
^%ZTSCH(“JOB”, run time, task #) = device $IO
^%ZTSCH(“LINK”)= “” or $H when dropped link was last checked
^%ZTSCH(“LINK”, volume set)= 1 if link has dropped
^%ZTSCH(“LINK”, volume set, next run time, task #)= “”
^%ZTSCH(“LOAD”, load rating) = cpu ^ $H when rating was checked
^%ZTSCH(“LOADA”, cpu) = whether TM should wait ^ load rating ^ $H
                                   ==>when rating was checked ^ $J of Manager
^%ZTSCH(“LOGRSRC”) = “”

^%ZTSCH(“NO-OPTION”)= “”
^%ZTSCH(“RUN”)= $H when Manager last checked in
^%ZTSCH(“STARTUP”, UCI, option #)= $H when option was first queued for
                               ==>startup
^%ZTSCH(“STATUS”, $J of Manager)= $H when Manager last checked in [1] ^
                               ==>status [2] ^ description of status [3]
^%ZTSCH(“STOP”)= “”
^%ZTSCH(“SUB”)= count of Submanagers waiting for tasks
^%ZTSCH(“TASK”, task #)= (A2) entry point [1] ^ (A3) routine [2] ^ (A4)
                      ==>option # [3] ^ (A5) option name [4] ^ (C6)
                      ==>description [5] ^ device name [6] ^ (E1) UCI [7] ^
                      ==>(C3) creation time [8] ^ (C1) creator DUZ or (C2)
                      ==>creator name [9] ^ $J of running task [10] ^ $H
                      ==>when task actually started running [11]
^%ZTSCH(“UPDATE”, $J of Manager)= $H when the Manager last updated its
                               ==>parameters
^%ZTSCH(“WAIT”)= “”
```

23.5.2 TASKS (#14.4) File

The **^%ZTSK** global holds this partially-VA FileMan-compatible file of tasks. It is structured with a descriptor node followed by sequential entries. The data dictionary for this file is 14.4, TASKS. It is a read-only file. The TASKS (#14.4) file has no cross-references, *not* even a top-level **B** cross-reference, and its descriptor node is updated by the **Queuable Task Log Cleanup** [XUTM QCLEAN] option.

Each entry itself contains a **zero** node and several decimal nodes followed by a number of storage nodes. Like the SCHEDULE file, the TASKS (#14.4) file is *not* available for direct manipulation or examination by application software. Site managers, however, can print out information on entries in the TASKS (#14.4) file using VA FileMan.

Figure 309 describes the nodes **0** through **.26** for each entry in the TASKS (#14.4) file:

Figure 309: TASKS (#14.4) File Nodes (1 of 2)

```
^%ZTSK(task #, 0)= (#.01) Entry Point [1F] ^ (#2) Routine Name [2F] ^ (#3) User
           ==>[3P:200] ^ (#4) Requested UCI [4F] ^ (#5) Creation Time ($H)
           ==>[5F] ^ (#6) Scheduled Run Time ($H) [6F] ^ (#7) Type of Task
           ==>[7F] ^ (#8) Option Number [8N] ^ (#9) Option Name [9F] ^ (#10)
           ==>Creator Name [10F] ^
           ==> (#11) Creation UCI [11F] ^ (#12) Creation Volume Set [12F] ^
           ==>(#13) RESERVED [13F] ^ (#14) Requested Volume Set [14F] ^ (#15)
           ==>Priority [15N] ^ (#16) Original Create date ($H) [16F]
^%ZTSK(task #, .01)= (#21) Original Destination UCI [1F] ^ (#22) Original
           ==>Destination Volume [2F] ^
^%ZTSK(task #, .02)= (#31) Current Destination UCI [1F] ^ (#32) Current
           ==>Destination Volume Set [2F] ^ (#33) Hop Count [3N] ^
^%ZTSK(task #,.03)= (#41) Task Description [E1,240F]^%ZTSK(D0,.04)= (#42) Schedule
Time Seconds [1N] ^
^%ZTSK(task #, .1)= (#51) Status Code [1F] ^ (#52) Last Update $H [2F] ^ (#53)
           ==>Status Notes [3F] ^ (#54) Job [4N] ^  ^  ^  ^ (#59.8) Remember
           ==>Until [8F] ^  ^ (#59.1) Stop Flag [10F]^
^%ZTSK(task #, .12, (#71) Error Count [1N] ^ (#72) Error $H [2F] ^ (#73) Error
           ==>Message [3F] ^
^%ZTSK(task #, .2)= (#81) Device IOP value [1F] ^ (#82) $IO value [2F] ^ (#83)
           ==>Device Type [3F] ^ (#84) Device Sub-Type [4F] ^ (#85) Device
           ==>%IS modifier [5F] ^ (#86) Host File Address [6F] ^ (#87) Sync Flag
[7F] ^ (#88) IO
            ==>Reschedule Count [8N] ^
^%ZTSK(task #, .21)= (D8) device file entry # [1] ^
^%ZTSK(task #, .25)= (D7) device parameters [1] ^
```

The remaining nodes of each entry are used to pass variables to the task. If the task has been manipulated only using TaskMan’s Program Interface, then the entries look like this:

Figure 310: TASKS (#14.4) File Nodes (2 of 2)

```
^%ZTSK(task #, .3, “name”)= (F2) value of saved variable
^%ZTSK(task #, .3, “array(“, node #)= (F2) value of saved variable
^%ZTSK(task #, .3, “array”, node #)= (F2) value of saved variable
```

The distinguishing characteristic here is the fact that the variables to be passed are all subscripted under the **.3**-node.

23.5.3 Task Status Codes

This section lists the various codes that may be found in the first ^-piece of the **.1** node, the text displayed for that code by the **List Tasks** action, and the meaning of that code. These codes are set into the tasks at every point in processing where the status changes, along with a time stamp and an explanation where necessary.

Several of the codes correspond to the status of the SCHEDULE file entry for the task. If all applications used the Program Interface, the status code would always agree with the task’s real status. In fact, many applications still directly manipulate **^%ZTSCH** and **^%ZTSK**, and they often neglect to update the status codes. Whenever the SCHEDULE file disagrees with the status code, the SCHEDULE file is correct. This is the reason many of the codes listed in Table 50 have multiple meanings.

Status codes **1** through **6** represent one of two common paths a task takes through TaskMan. The other common path replaces code **3** with **A**, where the task's device is *not* immediately available.

Table 50: TaskMan Task Status Codes

Status Code	Description
0	Incomplete or still being created.
1	Scheduled for <*date and time*>. TaskMan uses this status in every option and entry point that schedules a task. If the task fails or errors out and TaskMan *cannot* trap the error, this status has a different meaning: "Stopped irregularly while scheduled."
2	Being inspected by TaskMan. The Manager sets this status when the time comes for a task to run. As it removes the task from the SCHEDULE file, it sets this code into the task.
3	Waiting for a partition. When the Manager places a task in the Job list of the SCHEDULE file, it gives the task this code. If the task fails or errors out, and TaskMan *cannot* trap the error, this status has a different meaning: "Stopped irregularly while waiting for a partition."
4	Being prepared. The submanager gives a task this code when it removes the task from the Job list or Busy Device Waiting list in order to run it.
5	Currently running. The submanager gives a task this status just before it starts the task at its entry point. If the task fails or errors out, and TaskMan *cannot* trap the error, this status has a different meaning: "Started running <*date & time*> and stopped irregularly."
6	Completed <date and time>. The submanager gives a task this status after the task quits.
A	Waiting for device <*device name or* ***$I***>. The Manager or the submanager gives a task this status when it places the task in the Busy Device Waiting list. If the task fails or errors out and TaskMan *cannot* trap the error, this status has a different meaning: "Stopped irregularly while waiting for a device."
B	Rejected. <*rejection message*>. The Manager or the submanager gives a task this status if it fails one of the basic validation tests. (The rejection messages are contained in the "Task Rejection Messages" section.)

Status Code	Description
C	Error <*date and time*>. <*error message*>. The submanager gives a task this status if it traps an error after starting the task. The error message records the vendor-specific **$ZE** text.
D	Stopped by user. The manager or the submanager gives a task this status if, when TaskMan removes the task from the SCHEDULE file for processing, it finds that the user has asked the task to stop. The submanager also assigns this status if, just before starting the task, it finds the stop request has been made. Finally, the submanager gives a task this status if the task uses the **ZTSTOP** output variable to report that it stopped in response to a user's request. **REF:** For an explanation of **ZTSTOP**, see the description of $$S^%ZTLOAD API in the "TaskMan: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet website.
E	Interrupted while running. At startup, the Manager gives this status to any task listed in the Task list of the SCHEDULE file as still running.
F	Unscheduled by <*user name or "you"*>. The **Dequeue Tasks** [XUTM DQ] and **TaskMan User** [XUTM USER] options and the DQ^%ZTLOAD entry point use this status for tasks they unschedule.
G	Waiting for the link to <volume set name> to be restored. The Manager uses this status for tasks that would have been transferred to a different TaskMan environment and deleted from this one, if the local area network link to the remote environment were functioning properly. If the task fails or errors out, and TaskMan *cannot* trap the error, this status has a different meaning: "Stopped irregularly while waiting for a link."
H	Edited without being scheduled. The **Requeue Tasks** [XUTM REQ] and **TaskMan User** [XUTM USER] options and the REQ^%ZTLOAD entry point use this status when edited tasks are *not* subsequently rescheduled.
I	Discarded by TaskMan because its record was incomplete. The Manager or the submanager uses this status for tasks listed in the SCHEDULE file that lack critical information in the corresponding TASKS (#14.4) file entries.
J	Currently being edited. This status has been set aside for possible use in future versions of TaskMan.

Status Code	Description
K	Created without being scheduled. The ^%ZTLOAD entry point uses this status for tasks when the application passes **ZTDTH="@"**. Kernel Toolkit utility ^%ZTMOVE uses this value for the tasks it creates to transfer routines between Volume Sets manually.
L	Preparing this task caused the submanager an error <*date and time*>. <*error msg*>. The submanager uses this status when it traps an error after claiming a task but before starting it. The Manager does *not* yet record a corresponding status for the analogous situation. Tasks that never start, that are left with a status of **2**, have usually caused the Manager an error while it tried to examine them.
M	Waiting for a partition on a Compute Server. The Manager gives a task this code when it places the task in the Compute Server Job List. If the task fails or errors out, and TaskMan *cannot* trap the error, this status has a different meaning: "Stopped irregularly while waiting for a partition on a Compute Server."

23.5.4 Task Rejection Messages

Under certain conditions TaskMan can avoid trapping obvious errors by checking the tasks themselves for internal consistency. Whenever it finds tasks with bad data, it rejects them. This involves unscheduling them, setting their status codes to **B**, and adding a brief explanatory message. These messages listed in Table 51 can help identify bugs in application queuing software, in the local system configuration, or in TaskMan itself.

Table 51: TaskMan Rejection Messages

TaskMan Rejection Message	Description
BAD DESTINATION UCI	The Manager rejects a task for this reason under three different conditions: • If the task is bound for the Manager's own Volume Set, whatever value has been passed for the destination UCI *must* be a valid UCI on the current Volume Set. If **^%ZOSF("UCICHECK")** rejects the UCI, TaskMan rejects the task. • If the task is bound for a different Volume Set and the destination UCI is *not* listed in the UCI ASSOCIATION (#14.6) file under that Volume Set, the UCI *must* be accepted as a valid UCI on the current Volume Set so TaskMan can use File #14.6 to determine where the task should run. If **^%ZOSF("UCICHECK")** rejects the UCI, TaskMan rejects the task. • If the task is bound for a different Volume Set and that Volume Set's link is down and its REPLACEMENT VOLUME SET is the current Volume Set, TaskMan rejects the task.
BAD DESTINATION VOLUME SET	Every task's destination Volume Set *must* be listed in the VOLUME SET (#14.5) file.
BAD IO DEVICE <$I>	If a port goes bad while many tasks wait for it in the Busy Device Waiting list, TaskMan traps an error whenever the port is tested for availability. When the submanager traps such an error, it rejects every task waiting for that device.
INVALID OUTPUT DEVICE	The Manager performs a lookup on the devices that tasks request. If the ^%ZIS call indicates that the device does *not* exist, TaskMan rejects the task.
INVALID ROUTINE NAME	If a task's entry point is in a **%**-routine, the Manager tests for that routine's existence in the library UCI. If the routine does *not* exist there, TaskMan rejects the task.
NO DESTINATION UCI	When older applications bypassed the Program Interface, they sometimes scheduled tasks without

TaskMan Rejection Message	Description
	specifying the destination UCI. The Manager rejects all such tasks.
NO LINK ACCESS TO VOLUME SET	If the VOLUME SET (#14.5) file entry for a task's destination Volume Set indicates there is no link access to that Volume Set, the task is rejected.
NO ROUTINE AT DESTINATION	If a task's entry point is in a *non*-**%**-routine, then the check for the routine's existence is done by the submanager prior to starting the task.

23.5.5 TaskMan State Messages

When the Manager does *not* run, all background processing grinds to a halt. For this reason, the Manager's condition is of vital importance to system managers. When problems are detected with background processing at a site, checking the Manager's condition should be the first step. The Manager periodically records its state in the Status List. The **Monitor TaskMan** [XUTM ZTMON] option displays this list near the top of the screen. The various states and their meanings are described in the sections that follow.

23.5.5.1 BALANCE State

The Manager lists itself in this state if other managers (that are processing the same files) appear to have more CPU capacity available than the current Manager. While in the **BALANCE** state, the Manager does *not* process any tasks or start any new submanagers. The manager removes itself from the **BALANCE** state when it appears to have at least as much CPU capacity as the active Manager. In general, when many managers are working out of the same TASKS (#14.4) and SCHEDULE files, most of them are in the **BALANCE** state at any given time, with only the one or two least loaded managers actually processing tasks.

REF: For more information about TaskMan load balancing, see the "Multiple TaskMan Managers and Load Balancing" section in the "TaskMan: System Management—Configuration" section.

23.5.5.2 ERROR State

The Manager lists itself in the **ERROR** state after trapping errors. On some systems the process of recording an error is slow, so the presence of a distinct state helps identify the source of delay to the system manager. A troubleshooter who sees this state for TaskMan should immediately check the TaskMan Error list to see what kind of error is being recorded. Because TaskMan's code is structured as a series of nested loops, it can very easily generate thousands of errors a day under certain conditions.

23.5.5.3 PAUSE State

The **PAUSE** state means that some external condition is preventing the Manager from processing tasks. The description always indicates the cause. While in the **PAUSE** state, the Manager waits until the problem is resolved, checking once every **60** seconds. Table 52 lists the pause states:

Table 52: TaskMan PAUSE States

PAUSE State	Description
The following required **^%ZOSF** nodes are undefined, <list of nodes>	When the Manager starts, restarts, or recovers from a trapped error, its first order of business is to drop through some setup code that checks TaskMan's environment. If any critical **^%ZOSF** nodes are missing, it enters a **PAUSE** state and waits until the system manager restores the nodes.
Required link to <volume set name> is down	The other key check in the setup code is to ensure that all Volume Sets listed in the VOLUME SET (#14.5) file as required can actually be reached. The Manager tests each required link and enters the **PAUSE** state if any tests cause an error. The Manager remains in the **PAUSE** state, periodically testing the links, until they are restored.
Logons Inhibited	When the system manager sets the INHIBIT LOGONS? field of the VOLUME SET (#14.5) file, TaskMan enters a **PAUSE** state and waits until the flag is cleared.
No Signons Allowed	The system manager can use the software switch to stop logons, which places TaskMan in the **PAUSE** state.

23.5.5.4 RUN State

The **RUN** state (see Table 53) indicates that the Manager is going about its business in a relatively normal manner, managing background tasks on your system.

Table 53: TaskMan RUN States

RUN State	Description
Start	The Manager sets this value before and after executing the setup code at system startup.
Setup	The Manager identifies when it executes the setup code to test its environment.

RUN State	Description
Restart	The Manager sets this value after executing the setup code during a restart.
Main Loop	This should be the Manager's usual state. This indicates the Manager is executing the main loop that checks the environment, processes the Schedule list, and performs idle loop activities when appropriate.
TaskMan Job Limit Reached	When the total number of processes on the Manager's CPU exceeds the TaskMan Job Limit given in the VOLUME SET (#14.5) file, the Manager can continue to process the Schedule list but *cannot* start any new submanagers.

23.5.5.5 WAIT State

While in the **WAIT** state, the Manager does *not*:

- React to changes in its environment.
- Process tasks.
- Enter **PAUSE** states.
- Stop after the **Stop Task Manager** [XUTM STOP] option has been used.

The following two options let you create or undo the **WAIT** state:

- Place Taskman in a WAIT State Option [XUTM WAIT]
- Remove Taskman from WAIT State Option [XUTM RUN]

TaskMan *cannot* enter this state on its own; it can only be initiated manually. This is essentially a tool for you to tightly control the processing of tasks on your machines. The description for this state always reads "**TaskMan Waiting**".

V. Kernel Installation and Distribution System

24 KIDS: System Management—Installations

Kernel Installation and Distribution System (KIDS) was introduced with Kernel 8.0. Previously, software was exported using a utility called **DIFROM** and installed by running **INIT** routines that the **DIFROM** utility created. KIDS is the replacement for **DIFROM**; it introduces significant revisions to the software distribution and installation processes. This section introduces KIDS and describes some of the changes to the software export process.

Table 54 lists the definitions that apply throughout the KIDS documentation:

Table 54: KIDS-related Terms and Definitions

Term	Definition
Transport Global	An exported software application, stored in a global. KIDS exports software (i.e., package) based on its definition in a build entry. The transport global also contains the build entry and the PACKAGE (#9.4) file entry (if any) for a given software application.
Build Entry	An entry in the BUILD (#9.6) file that defines the parts of a software application to export. Also known as a build.
Component	An element of one of the following types: • Template (PRINT, SORT, and INPUT) • Form • Function • Bulletin • Help Frame • Routine • Option • Security Key • Protocol
Distribution	A Host File Server (HFS) file containing transport globals. If a distribution contains multiple transport globals, KIDS treats them as a single installation when installing from the distribution.
Package	A cohesive set of files, data, and components that together form a set of computing activities related to a functional area (i.e., software).

24.1 KIDS Options

To get to the KIDS: **Kernel Installation & Distribution System** [XPD MAIN] menu (locked with the XUPROG security key) choose the **Programmer Options** [XUPROG] menu option on the **Kernel Systems Manager Menu** [EVE], as shown in Figure 311:

Figure 311: KIDS Menu Options

```
Select Systems Manager Menu Option: PROGRAMMER OPTIONS


KIDS   Kernel Installation & Distribution System ...                      [XPD MAIN]
             **> Locked with XUPROG
   PG     Programmer mode                                               [XUPROGMODE]
             **> Locked with XUPROGMODE
          Delete Unreferenced Options                           [XQ UNREF'D OPTIONS]
          Error Processing ...                                              [XUERRS]
          General Parameter Tools ...                              [XPAR MENU TOOLS]
          Global Block Count                                        [XU BLOCK COUNT]
          List Global                                                       [XUPRGL]
             **> Locked with XUPROGMODE
          Routine Tools ...                                     [XUPR-ROUTINE-TOOLS]
          Test an option not in your menu                           [XT-OPTION TEST]
             **> Locked with XUMGR
Select Programmer Options Option: KIDS <Enter> Kernel Installation & Distribution
    System


          Edits and Distribution ...                         [XPD DISTRIBUTION MENU]
          Utilities ...                                                [XPD UTILITY]
          Installation ...                                   [XPD INSTALLATION MENU]
             **> Locked with XUPROGMODE
          Patch Monitor Main Menu ...                 [XTPM PATCH MONITOR MAIN MENU]
          Patchman ...                                 [XPD AUTOMATIC PATCHING MENU]
```

As indicated by its name (i.e., KIDS = Kernel Installation and Distribution System), KIDS supports two major functions:

- Distributions
- Installations

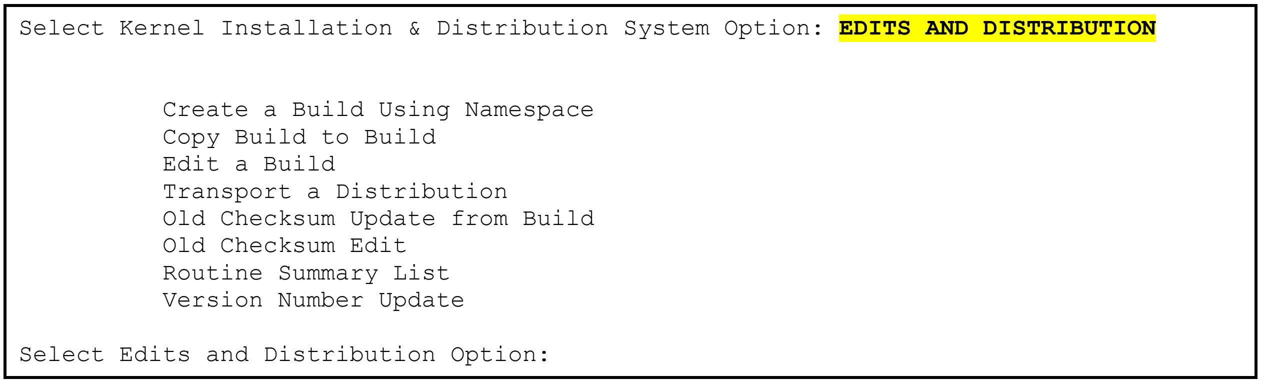
REF: In addition, KIDS also provides other utilities. For more information on KIDS utilities, see the “KIDS: System Management—Utilities” section.

24.1.1 Distributions

The distribution related options are located on the **Edits and Distribution** [XPD DISTRIBUTION MENU] menu (see Figure 312). The distribution portion of KIDS allows developers to:

- Define the contents of a software application in a build entry.
- Create transport globals from build entries.
- Export transport globals by creating distributions.

Figure 312: Edits and Distribution Menu Options



```
Select Kernel Installation & Distribution System Option: EDITS AND DISTRIBUTION


          Create a Build Using Namespace
          Copy Build to Build
          Edit a Build
          Transport a Distribution
          Old Checksum Update from Build
          Old Checksum Edit
          Routine Summary List
          Version Number Update

Select Edits and Distribution Option:
```

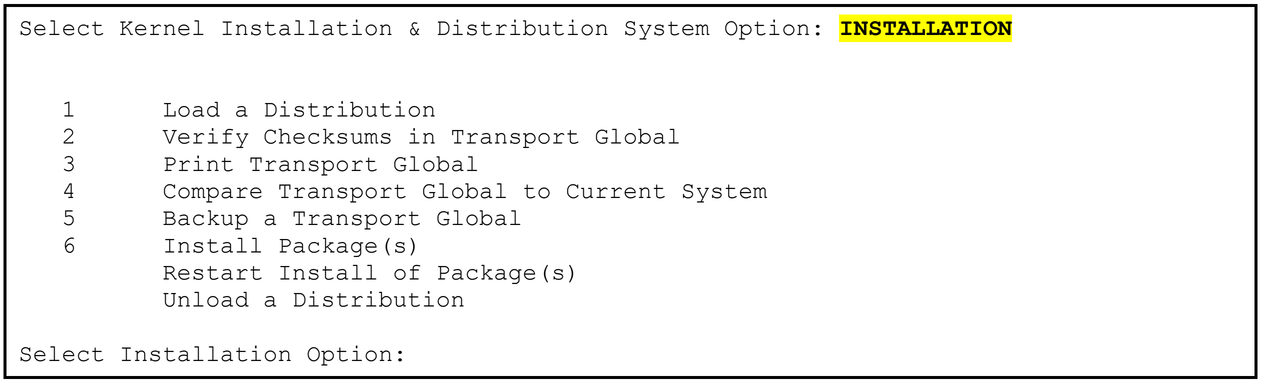
REF: For a description on how application developers use the KIDS build and distribution options, see the "KIDS: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

24.1.2 Installations

The installation related options are located on the **Installation** [XPD INSTALLATION MENU] menu (see Figure 313). The installation portion of KIDS allows sites to:

- Load transport globals from KIDS distributions.
- Load transport globals from KIDS PackMan messages.
- Print out the contents of loaded transport globals before installing them.
- Compare the contents of loaded transport globals to the current system before installing them.
- Install loaded transport globals.

Figure 313: Installation Menu Options



```
Select Kernel Installation & Distribution System Option: INSTALLATION


   1      Load a Distribution
   2      Verify Checksums in Transport Global
   3      Print Transport Global
   4      Compare Transport Global to Current System
   5      Backup a Transport Global
   6      Install Package(s)
          Restart Install of Package(s)
          Unload a Distribution

Select Installation Option:
```

KIDS introduced two files into Kernel:

- BUILD (#9.6) file
- INSTALL (#9.7) file

KIDS also makes use of the existing PACKAGE (#9.4) file, but its role in exporting and installing software is diminished.

24.2 Build Entries and the BUILD (#9.6) File

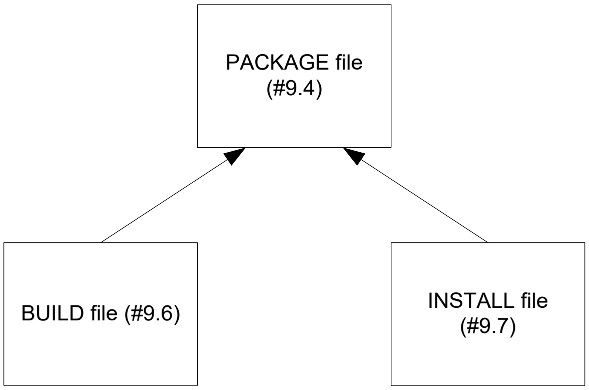
Build entries, stored in the BUILD (#9.6) file, are where developers define a software application. This build entry defines the set of files, data, components, installation questions, national software information, pre- and post-install routines, and other settings that comprise the exported software.

Software components are no longer tied to namespace, as they were previously with DIFROM and the PACKAGE (#9.4) file. Developers can select any components available on the current system and include them in their build entries as software components.

The format of the NAME (#.01) field of a build entry *must* be the software name concatenated with a space, and then a version number. This means that there is a separate entry for every version of a software application that a developer exports.

Also, a software application's build entry is sent to installing sites as part of the software; after an installation, the site can examine the build entry to see the software definition.

Figure 314: KIDS File Diagram



24.3 INSTALL (#9.7) File

The INSTALL (#9.7) file stores a record of each installation a site performs. The INSTALL (#9.7) file allows KIDS to store a separate installation entry for each installation. A new version of software no longer overwrites the installation information of a previous version, and developers' installation history no longer overwrites the sites' installation history. The national PACKAGE (#9.4) file is now static at its top level.

The three main items recorded in the INSTALL (#9.7) file for each installation are the installing site's answers to installation questions, any installation output, and the installation's timing information.

24.4 Changes in the Role of the PACKAGE (#9.4) File

The PACKAGE (#9.4) file still plays a role in installations with KIDS, albeit a diminished one. KIDS provides a link from the build entry of a package to the PACKAGE file, so that developers can link a package to a PACKAGE (#9.4) file entry.

The top level of a PACKAGE (#9.4) file entry for a package now stores static package information. The only part of the PACKAGE (#9.4) file entry that installations update automatically now is the VERSION Multiple field. A patch sent with KIDS does *not* transport the entire PACKAGE (#9.4) file entry. It only sends the information that is needed to update the PACKAGE (#9.4) file. Patch installations updates the PATCH APPLICATION HISTORY Multiple field, which is within the VERSION Multiple field. KIDS saves patch names along with their sequence numbers in this multiple. Most other fields have been designated for removal at the top level of the PACKAGE (#9.4) file. The PACKAGE (#9.4) file now stores mainly static software information that is *not* version specific, as well as the patch history of the software.

24.5 Transport Mechanism: Distributions

Distributions are the mechanism KIDS uses to export software. They are more flexible than the previous mechanism (**INIT** routines).

Distributions are usually in the form of an HFS file. The developer creates transport globals from build entries. KIDS stores transport globals in a global. KIDS can **WRITE** the global (in a format readable only by KIDS) to an HFS file; the HFS file is the distribution. The HFS file can then be distributed by a variety of methods, including FTP (file transfer protocol), diskette, and tape. For example, if your system is a PC, you can also move the Transport Global to a new medium (i.e., to multiple floppy disks so you can install on other PCs):

1. Select the **Load a Distribution** [XPD LOAD DISTRIBUTION] option (*Do not* run the Environment Check routine).
2. Under the **Utilities** [XPD UTILITY] menu, select the **Convert Loaded Package for Redistribution** [XPD CONVERT PACKAGE] option.
3. Under the **Edits and Distribution** [XPD DISTRIBUTION MENU] menu, select the **Transport a Distribution** [XPD TRANSPORT PACKAGE] option.
4. At the “Enter a Host File:” prompt, enter the floppy drive and file name. For example:

```
Enter a Host File: A:\KRN8.KID)
```

One advantage to using distributions over **INIT** routines is that there is no limit to the size of a software application you can export. Another advantage is that during installations, you no longer have to overwrite a software application’s existing routines with the new routines before running the installation.

Alternatively, a KIDS distribution can be sent via a PackMan message in MailMan. But transporting software as host files, especially large ones, avoids slowing down MailMan.

24.5.1 Two Kinds of Distributions

KIDS supports two kinds of distributions:

- **Standard Distribution—**This type of distribution contains transport globals for what are traditionally thought of as software applications, including files, data, and all components. A standard distribution can contain one or more transport globals. If there is more than one transport global, KIDS treats each one as a single installation unit.
- **Global Distribution—**This type of distribution contains one transport global only, and that transport global can export M globals only.

The transport globals in both types of distributions also contain the corresponding build entry, and if linked to a PACKAGE [#9.4] file entry, the corresponding PACKAGE (#9.4) file entry. However, a patch sent with KIDS does *not* transport the entire PACKAGE (#9.4) file entry. It only sends the information that is needed to update the PACKAGE (#9.4) file.

24.6 What Happens to DIFROM?

Developers should no longer use the DIFROM entry point to export software. Developers should use KIDS. The DIFROM method is still supported, but only for the support of sites that use standalone VA FileMan (VA FileMan without Kernel).

REF: For more information on using DIFROM, see the *VA FileMan Programmer Manual*.

24.7 Installing Standard Distributions

As noted previously, KIDS supports two types of distributions:

- Standard
- Global

This section describes how KIDS installations work when installing standard distributions.

24.7.1 Installation Sequence

KIDS installs standard distributions in three phases:

1. Loading transport globals from the distribution.
2. Answering installation questions for each transport global.
3. Installing each transport global in the distribution.

24.7.1.1 Phase 1: Loading Transport Globals from a Distribution or PackMan Message

1. Using the **Load a Distribution** [XPD LOAD DISTRIBUTION] option, the installer chooses the HFS file from which to load distributions. If loading from a PackMan message, choose the message and invoke MailMan's **INSTALL/CHECK MESSAGE** option under the **Load PackMan Message** [XMPACK] option.
2. For each transport global, KIDS makes an entry in the INSTALL (#9.7) file for the transport global.
3. KIDS loads transport globals from distribution into **^XTMP**.
4. KIDS prompts the user to see if they want to run the environment check for each transport global (if unsuccessful, the process quits here; the developer may or may *not* **KILL** INSTALL (#9.7) file entries and transport globals from **^XTMP**.)
5. The installer can print the contents of the transport global, compare the contents to the current system, and verify checksums of the transport global.

24.7.1.2 Phase 2: Answering Installation Questions for Transport Globals in a Distribution

1. Using the **Install Package(s)** [XPD INSTALL BUILD] option, the installer selects a distribution to install by choosing an entry from the INSTALL (#9.7) file.
2. KIDS runs the environment check for the first transport global; the environment check can allow KIDS to install the transport global, cancel installation of the transport global, or cancel installation of all transport globals in the distribution.
3. The installer answers pre-installation questions for the first transport global.
4. The installer answers standard KIDS questions for the first transport global.
5. The installer answers post-installation questions for the first transport global.
6. The installer repeats Steps #2-5 for the remaining transport globals, if there are any more transport globals to process.
7. The installer chooses a device for the installation to run on. The installer can queue the installation or run it directly; entering a caret (^) aborts the installation.

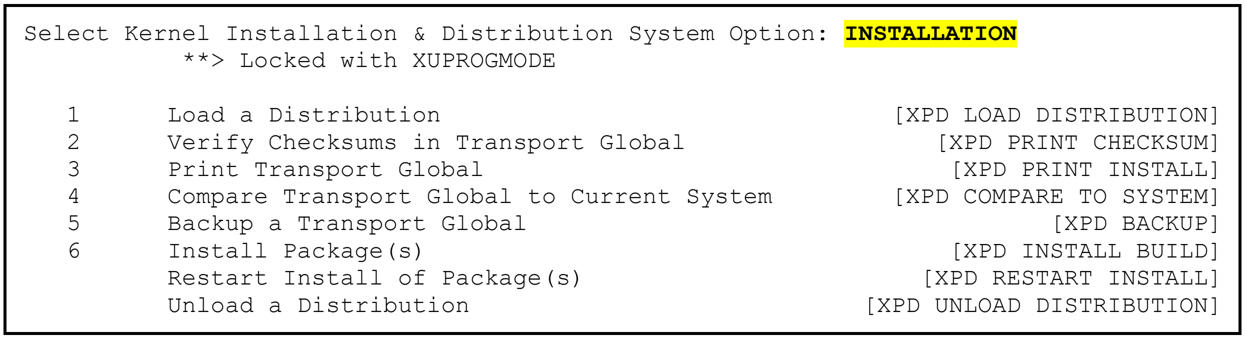
24.7.1.3 Phase 3: KIDS Installation of Software

1. KIDS disables any options and protocols the site has asked to be disabled for this install. However, KIDS does *not* disable options and protocols which have an Action of USE AS LINK FOR MENU ITEMS.
2. KIDS waits for the time period (from **0** to **60** minutes) the site specifies, if they chose to disable options and protocols.
3. KIDS suspends the running of queued options by TaskMan for this install, if the site chooses to do so.
4. The pre-install routine is run for the first transport global.
5. All components are installed for the first transport global.
6. The post-install routine is run for the first transport global.
7. KIDS repeats Steps 4-6 for any remaining transport globals to install in the distribution.
8. Options and protocols that were disabled for this install (if any) are re-enabled.
9. Queued options are removed from suspense (if the site chose to suspend queued options).

24.7.2 Installation Menu

The KIDS **Installation** [XPD INSTALLATION MENU] menu contains the options shown in Figure 315:

Figure 315: KIDS Installation Menu Options



```
Select Kernel Installation & Distribution System Option: INSTALLATION
           **> Locked with XUPROGMODE

   1      Load a Distribution                              [XPD LOAD DISTRIBUTION]
   2      Verify Checksums in Transport Global                [XPD PRINT CHECKSUM]
   3      Print Transport Global                               [XPD PRINT INSTALL]
   4      Compare Transport Global to Current System       [XPD COMPARE TO SYSTEM]
   5      Backup a Transport Global                                   [XPD BACKUP]
   6      Install Package(s)                                   [XPD INSTALL BUILD]
          Restart Install of Package(s)                      [XPD RESTART INSTALL]
          Unload a Distribution                          [XPD UNLOAD DISTRIBUTION]
```

The number next to the options indicates the order of the option entries you should follow when performing a KIDS installation.

24.7.3 Loading a Standard Distribution

The first step in installing a standard distribution is to load the transport globals from the distribution. The **Load a Distribution** [XPD LOAD DISTRIBUTION] option does the following:

- Lists what transport globals are contained in the distribution and asks you if you want to continue.
- Creates entries in the INSTALL (#9.7) file for each transport global in the distribution that passed its environment check.
- Loads transport globals from the distribution (HFS file) into the **^XTMP** global (if you answer **YES** to continue).
- Prompts the user to see if they want to run the environment check for each transport global. If a transport global does *not* pass its environment check, KIDS may purge it from **^XTMP**; otherwise, the transport global stays in **^XTMP**. KIDS tells you the result of each environment check.
- Checks the version number of the incoming software against any existing software of the same name at the site. If the incoming version number is *not* greater than the existing version, KIDS aborts the installation for the transport global in question.
- Echoes the name of the first transport global to pass environment check (i.e., “Use transport global name to install this Distribution”). The name of the first transport global to pass its environment check is the name you use to install the distribution, in the next phase.

Loading a distribution is the first of three phases to install VistA software. The second phase is answering installation questions, including scheduling the installation; the third and final phase is the actual running of the installation.

When loading from a PackMan message, load the distribution using MailMan's **INSTALL/CHECK MESSAGE** option under the **Load PackMan Message** [XMPACK] option. For KIDS PackMan messages, this option through MailMan is equivalent to the **Load a Distribution** [XPD LOAD DISTRIBUTION] option.

Figure 316: Load a Distribution Option—Sample User Dialog

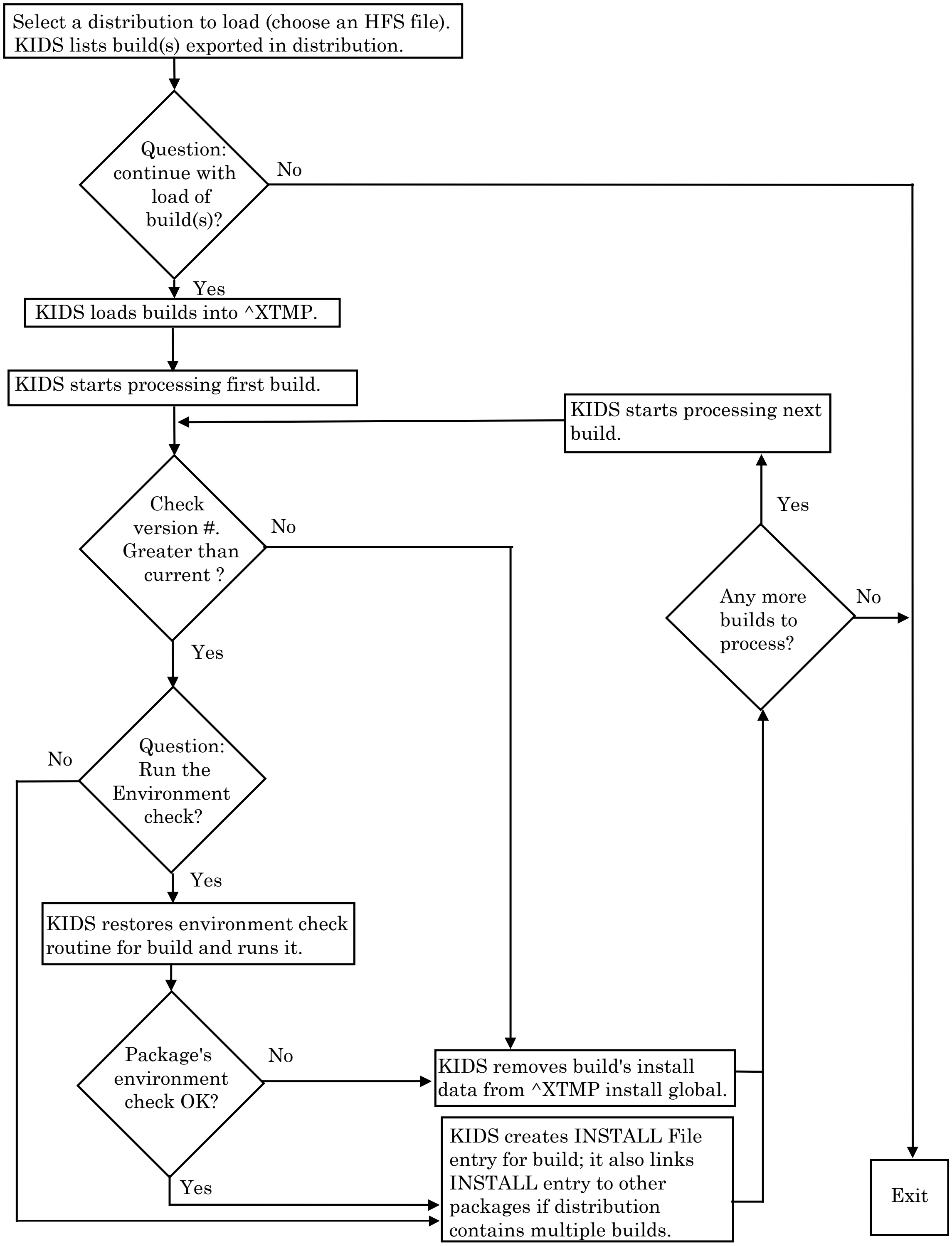
```
Select Installation Option: LOAD A DISTRIBUTION
Enter a Host File: ZXG_EXPT.DAT

Distribution saved on Oct 13, 2004@09:29:08
Comment: TEST PKGS

This Distribution contains Transport Globals for the following Package(s):
     TEST 2.1

Want to Continue with Load? YES// <Enter>
Loading Distribution...

Want to RUN the Environment Check Routine? YES// <Enter>
   TEST 2.1

Use INSTALL NAME: TEST 2.1 to install this Distribution.

Select Installation Option:
```

24.7.3.1 When the Distribution is Split across Diskettes

Distributions can come in a single host file (see Figure 312); alternatively, they can come on diskettes, with the host file split up among the diskettes. If you are installing from a distribution that is spread across diskettes, the **Load a Distribution** [XPD LOAD DISTRIBUTION] option asks you for subsequent diskettes (e.g., "Insert the next diskette, #2, and Press the return key", etc.). Insert the appropriate disk, press the **<Enter>** key, and continue until the distribution is loaded.

24.7.4 Loading Transport Globals from a Distribution

Figure 317: Loading Transport Globals from a Distribution—Flowchart


Select a distribution to load (choose an HFS file). KIDS lists build(s) exported in distribution.
Question: continue with load of build(s)?
No
Yes
KIDS loads builds into ^XTMP.
KIDS starts processing first build.
KIDS starts processing next build.
Yes
Check version #. Greater than current ?
No
Any more builds to process?
No
Yes
No
Question: Run the Environment check?
Yes
KIDS restores environment check routine for build and runs it.
Package's environment check OK?
No
KIDS removes build's install data from ^XTMP install global.
KIDS creates INSTALL File entry for build; it also links INSTALL entry to other packages if distribution contains multiple builds.
Yes
Exit

24.7.5 Verifying Checksums in a Transport Global

You can verify the checksums for a loaded transport global in advance of installing from it, using the **Verify Checksums in Transport Global** [XPD PRINT CHECKSUM] option. This option verifies all checksums of routines in the transport global, reporting any discrepancies. In the future, the ability to verify checksums will be extended to other KIDS components besides routines.

As of Kernel patch XU*8.0*369, the integrity checking CHECK1^XTSUMBLD routine supports the **Compare local/national checksums report** [XU CHECKSUM REPORT] option.

As of Kernel patch XU*8.0*393, KIDS was modified to send a message to a server on FORUM when a KIDS build is sent to a Host File Server (HFS) device. This message contains the checksums for the routines in the patch. The server on FORUM matches the message with a patch if the sending domain is authorized on FORUM. There is no longer a need for developers to manually include routine checksums (either CHECK^XTSUMBLD or CHECK1^XTSUMBLD routines) in the patch description. The patch module includes the before and after CHECK1^XTSUMBLD values in the Routine Information section at the end of the patch document.

With changes in the National Patch Module (NPM) on FORUM, when the patch is released the checksums for the routines are moved to the ROUTINE (#9.8) file on FORUM. The checksum "before" values come from the FORUM ROUTINE (#9.8) file and are considered the GOLD standard for released checksums. The local site's **Compare local/national checksums report** [XU CHECKSUM REPORT] option uses the FORUM ROUTINE (#9.8) file as its source to create reports showing any routines that do *not* match.

This patch also modified the KIDS BUILD (#9.6) file by adding the TRANSPORT BUILD NUMBER (#63) field used to store a build number that is incremented each time a build is made. This build number is added to the second line of each routine in the **7th** "**;**" piece. This makes it easy to tell if a site is running the current release during testing and afterword. The leading "B" found in the checksum tells the code what checksum routine to use.

24.7.6 Printing Loaded Transport Globals

Once you have loaded transport globals from a standard distribution onto your system, you can print out the definitions of the transport globals, using the **Print Transport Global** [XPD PRINT INSTALL] option. This way, you can see every component exported in each transport global, before you install them.

Figure 318: Print Transport Global Option—Sample Printed Transport Global

```
PACKAGE: ZXG DEMO 1.0                                                   PAGE 1
----------------------------------------------------------------------------
NATIONAL PACKAGE:
DESCRIPTION:

ENVIRONMENT CHECK : ZXGENV
PRE-INIT ROUTINE : ZXGPRE
POST-INIT ROUTINE: ZXGPOS
----------------------------------------------------------------------------

ROUTINE:
   ZXGC00                                        SEND TO SITE
   ZXGC01                                        SEND TO SITE
   ZXGC02                                        SEND TO SITE
   ZXGCMOVE                                      SEND TO SITE
   ZXGCTEST                                      SEND TO SITE
   ZXGCTW1                                       SEND TO SITE
   ZXGCWE                                        SEND TO SITE
   ZXGCXMP1                                      SEND TO SITE
   ZXGCXMPL                                      SEND TO SITE
   ZXGDEMO                                       SEND TO SITE
   ZXGKC                                         SEND TO SITE
   ZXGLMSG                                       SEND TO SITE
   ZXGLOAD                                       SEND TO SITE
   ZXGTMP                                        SEND TO SITE

INSTALL QUESTIONS:
     SUBSCRIPT: PRE1
DIR(0)=YA^^
DIR(“A”)=Do you want to run the pre-install conversion?
DIR(“B”)=YES
DIR(“?”)=Answer YES to run the pre-install conversion, NO to skip it...
```

24.7.7 Comparing Loaded Transport Globals to the Current System

When you have loaded transport globals from a standard distribution onto your system, you can also compare a transport global to the matching software already installed on your system (if any), using the **Compare Transport Global to Current System** [XPD COMPARE TO SYSTEM] option. This way, you can compare the software you are about to install with the current version of the software on your system.

When this option finds differences, it notes the change by displaying the differences between the current software and the transport global on two lines, one line labeled *** OLD *** and the other *** NEW ***.

NOTE: Pointers are converted to FREE TEXT when exporting VA FileMan entries, so these converted free pointers show up as differences when using the compare feature.

Figure 319: Compare Transport Global to Current System Option—Sample Comparison Output

```
Compare ZXP 1.0 to current site
-------------------------------------------------------------------


 Routine: ZUVXD


 File # 3.2 Data Dictionary


 File # 3.2 Data

* OLD *   ^%ZIS(2,9,8) =
$C(27)_”[A”^$C(27)_”[B”^$C(27)_”[C”^$C(27)_”[D”^3^^$C(27)_”[L”
* NEW *   ^%ZIS(2,9,8) = $C(27)_”[A”^$C(27)_”[B”^$C(27)_”[C”^$C(27)_”[D”^3
* OLD *   ^%ZIS(2,44,13) = ^$C(26)^^^^$J(“”,X)_$C(27,93,($X+32-X))
* NEW *   ^%ZIS(2,44,13) = ^$C(26)^^^^
* OLD *   ^%ZIS(2,60,8) =
$C(27)_”[A”^$C(27)_”[B”^$C(27)_”[C”^$C(27)_”[D”^3^^$C(27)_”[L”
* NEW *   ^%ZIS(2,60,8) = $C(27)_”[A”^$C(27)_”[B”^$C(27)_”[C”^$C(27)_”[D”^3
* ADD *   ^%ZIS(2,93,21) = ^


HELP FRAME


BULLETIN
```

This option was updated with Kernel patch XU*8.0*393 to add a side-by-side comparison in columnar format, which only works if Kernel Toolkit patch XT*7.3*93 has also been installed, as shown in Figure 320:

Figure 320: Compare Transport Global to Current System Option—Sample Comparison Output in Columnar Format

```
Select Kernel Installation & Distribution System Option:

   1      Load a Distribution
   2      Verify Checksums in Transport Global
   3      Print Transport Global
   4      Compare Transport Global to Current System
   5      Backup a Transport Global
   6      Install Package(s)
          Restart Install of Package(s)
          Unload a Distribution

Select Installation Option: 4 <Enter> Compare Transport Global to Current System
Select INSTALL NAME: XU*8.0*381 <Enter> Loaded from Distribution
Loaded from Distribution  9/14/06@12:39:52
     => DEMO COMPARE  ;Created on Sep 14, 2006@12:39:17

This Distribution was loaded on Sep 14, 2006@12:39:52 with header of
   DEMO COMPARE  ;Created on Sep 14, 2006@12:39:17
   It consisted of the following Install(s): XU*8.0*381

     Select one of the following:

          1         Full Comparison
          2         Second line of Routines only
          3         Routines only
          4         Columnar Routine compare

Type of Compare: 4 <Enter> Columnar Routine compare
DEVICE: HOME// <Enter>  Telnet terminal

Compare XU*8.0*381 to current site     Routines Only
--------------------------------------------------------------------------

Compare of routines from KIDS XU*8.0*381, and disk

Routine XU8P381 not on disk
--------------------------------------------------------------------------
Routine XUTMTP
   KIDS                                    Disk
--------------------------------------------------------------------------
1{XUTMTP ;SEA/RDS - TaskMan:ToolKit} 1{XUTMTP ;SEA/RDS - TaskMan: ToolKit}
 {, Print, Part 1 ;04/18/2006  16:19} {, Print, Part 1 ;04/24/2003  11:06}
                        ^                                       ^
2{ ;;8.0;KERNEL;**20,86,169,242,381*}2{ ;;8.0;KERNEL;**20,86,169,242**;Ju}
                                ^                                       ^
--------------------------------------------------------------------------
```

24.7.8 Backing Up Transport Globals

The **Backup a Transport Global** [XPD BACKUP] option allows the user to back up just the routines or create a backup build of all parts of a patch. It copies all of the existing files, fields, routines, and components that are updated by the patch into a backup build. The end result is a MailMan message or a Host file that can be installed and restores your system back to a state before the patch.

Figure 321: Backup a Transport Global——Sample System Prompts and User Entries

```
Select Installation <TEST ACCOUNT> Option: BACKUP A TRANSPORT GLOBAL
Select INSTALL NAME: XT*7.3*147 <Enter> Loaded from Distribution    7/27/20@13:32
:07
     => XT*7.3*147 TEST v1

This Distribution was loaded on Jul 27, 2020@13:32:07 with header of
   XT*7.3*147 TEST v1
   It consisted of the following Install(s):
     XT*7.3*147

Subject: Backup of XT*7.3*147 on Feb 12, 2021  Replace

     Select one of the following:

          B         Build (including Routines)
          R         Routines Only

Backup Type: B// ?

Backup the entire Build(routines, files, options, protocols, templates,
etc.) or just the Routines.

     Select one of the following:

          B         Build (including Routines)
          R         Routines Only

Backup Type: B// ??

Enter 'B' to create a backup of this Build. A new Build will be created using
the same Build name with a 'b' appended to the end. This new Build will be used
to create a KIDS backup of routines, files, options, protocols, templates, etc.
If this backup is a single build, a Packman email is created.  If it is a multi-
package a Host File is created.
Enter 'R' to create a Packman email of only the routines.

     Select one of the following:

          B         Build (including Routines)
          R         Routines Only

Backup Type: B// <Enter> uild (including Routines)

Created by XUUSER,ONE at <REDACTED>.VA.GOV  (KIDS) on Friday,
02/12/21 at 08:43
Do you wish to secure this message? NO// <Enter>
Send mail to: XUUSER,ONE // <Enter> XUUSER,ONE
Select basket to send to: IN// <Enter>
And Send to: <Enter>
```

```
Message sent
```

24.7.9 Running Installations

Once you have loaded the transport globals from a standard distribution, you can install them. Do this using the **Install Package(s)** [XPD INSTALL BUILD] option.

When you load a distribution, KIDS tells you which transport global name to use to install the distribution (e.g., “Use PACKAGE 1.0 to install this Distribution”). This is always the first transport global to successfully load from the distribution. When you use the **Install Package(s)** [XPD INSTALL BUILD] option, select the transport global name reported when you loaded the original distribution. Once you’ve done that, you can answer the installation questions for each transport global in the distribution.

24.7.9.1 Processing Each Transport Global

When you select a distribution to install, the **Install Package(s)** [XPD INSTALL BUILD] option processes the installation questions for each transport global in the distribution. For each transport global, you are asked:

- Pre-Install questions.
- Standard KIDS Questions.
- Post-Install Questions.
- Whether to disable any options or protocols. By typing three question marks (**???**) at this prompt KIDS lists all of the options and protocols it will disable. If you answer **YES**, all incoming options and protocols are disabled. You are also prompted to add to or delete from the list of options and protocols to disable. However, KIDS does *not* disable options and protocols which have an Action of USE AS LINK FOR MENU ITEMS. All scheduled options on the system are also disabled. Finally, you are asked a time period for installation:

  ```
  Delay Install(Minutes): (0-60): 0//”
  ```

  You can delay before starting the installation after disabling options and protocols from 0 to **60** minutes. This is to allow users already in (disabled) options time to exit the options before the installation starts.

24.7.9.2 Scheduling Installations

The final question you are asked when using the **Install Package(s)** [XPD INSTALL BUILD] option to load software is upon what device to run the installation. Your choices at the “DEVICE:” prompt are:

- Run the installation directly by selecting a device without queueing. The installation runs immediately, on the device you specify.
- Queue the installation.

- Abort the installation of the distribution by entering a caret (^).

24.7.10 When the Installation is Queued

If you queued the installation, you can look up the installation task in TaskMan. A KIDS installation task looks like Figure 322:

Figure 322: Queued KIDS Installation—Sample Installation Task

```
3: (Task #1179950) EN^XPDIJ, KIDS install.  Device VER$LW.  KRN,KDE.
   From TODAY at 16:24,  By you.  Scheduled for TODAY at 22:00
```

You can cancel a queued installation (before it has started) by deleting the task. KIDS also allows you to restart an install if the install is queued and you get an error during the installation.

24.7.11 Re-answering Installation Questions

If you queued an installation, you can re-answer installation questions, if you so choose, using the **Install Package(s)** [XPD INSTALL BUILD] option. To be able to re-answer the questions, however, you need to locate the task that was queued for the installation and delete it first. Once you delete the installation's queued task, you can re-answer the install questions. When you re-answer questions, your answers from the previous time come up as default responses.

Also, if you abort an installation after answering its installation questions (i.e., by entering a caret [^]), your responses are again used as the defaults the next time you try to install.

24.7.12 Information Stored in the INSTALL (#9.7) File

KIDS exports the definition of a software application in the BUILD (#9.6) file. KIDS records installations of software in the INSTALL (#9.7) file. The installation records in the INSTALL (#9.7) file provide a record of the start time, timing for each checkpoint, and completion time (if any) for an installation.

When an installation aborts, the contents of the INSTALL (#9.7) file determine where the install starts up again when you use the **Restart Install of Package(s)** [XPD RESTART INSTALL] option (checkpoint information is stored in the INSTALL [#9.7] file).

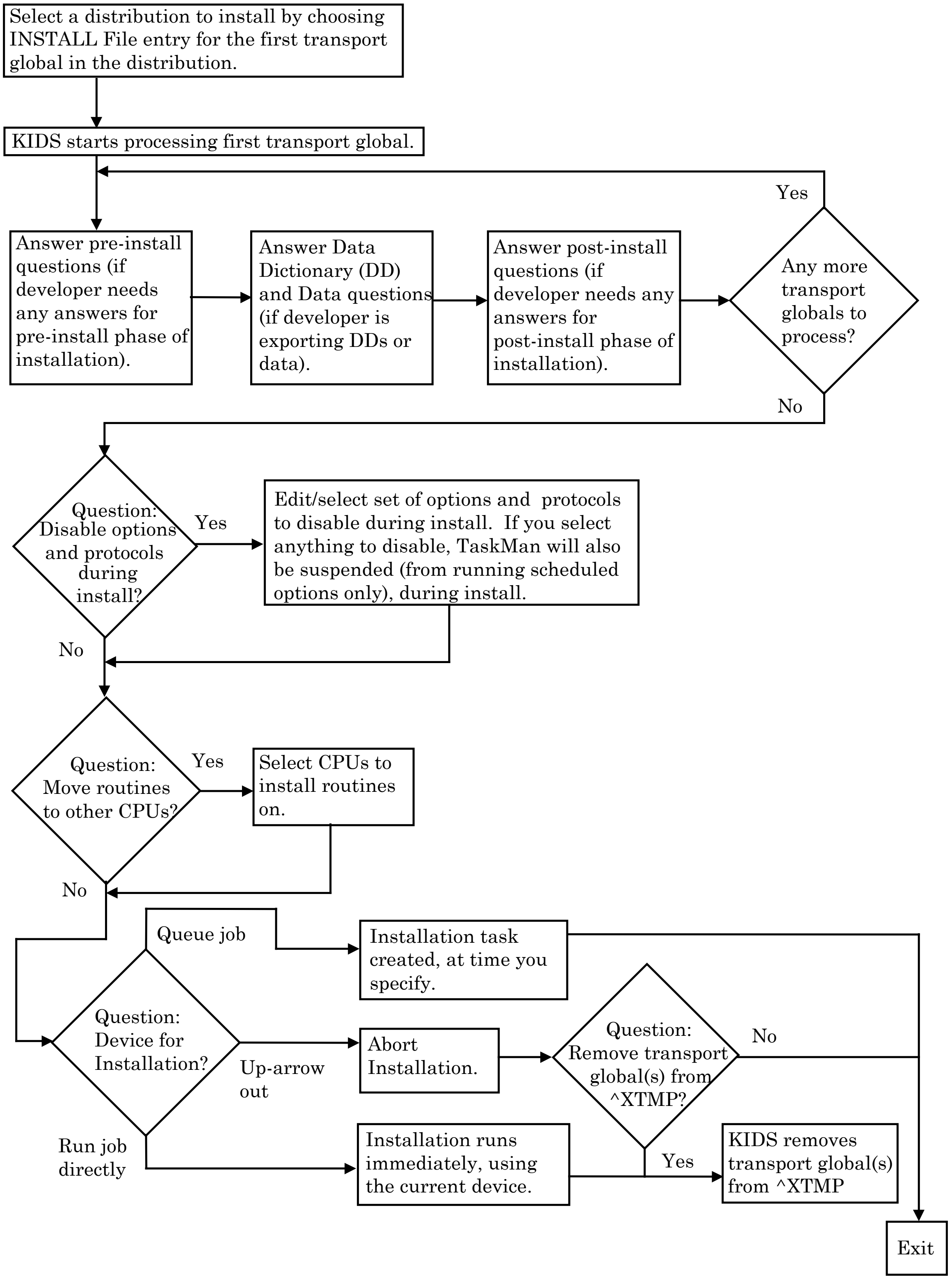
As well as being sent to the installation's principal device, all output from the installation is also stored in the INSTALL (#9.7) file, in the MESSAGES word-processing-type field.

The installation questions (and your answers to them) are stored in the INSTALL ANSWERS (#50) Multiple field of the INSTALL (#9.7) file.

You can print entries from the INSTALL (#9.7) file with the **Install File Print** [XPD PRINT] option.

24.7.13 Answering Installation Questions for a Distribution

Figure 323: Answering Installation Questions for a Distribution—Flowchart

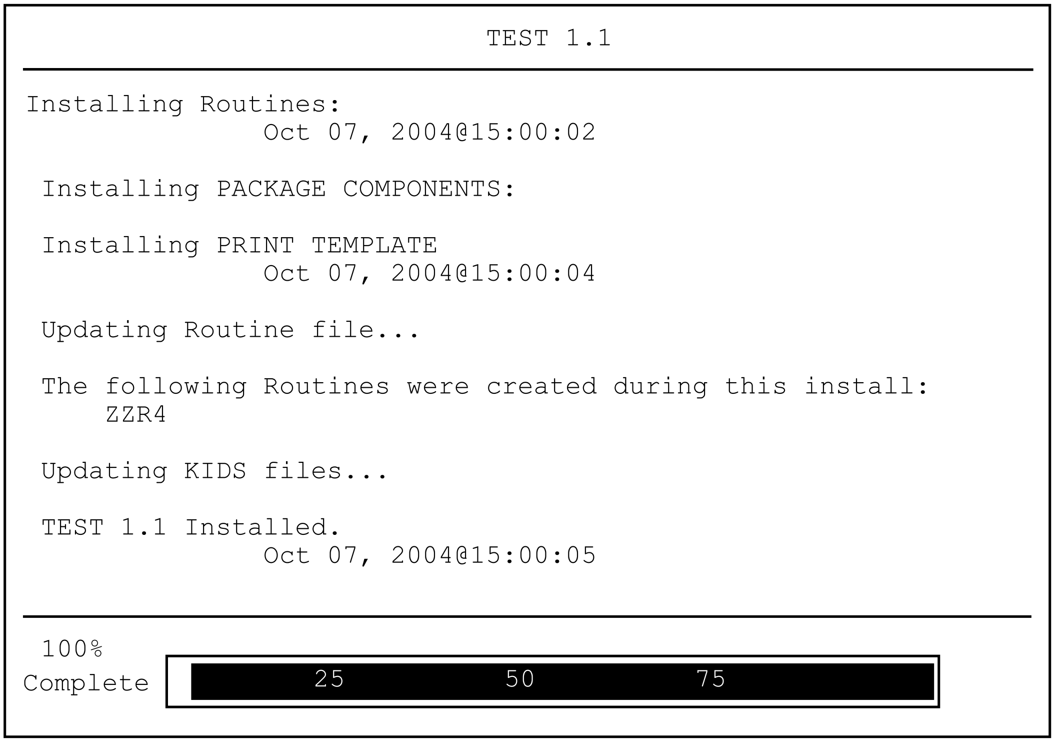

Select a distribution to install by choosing INSTALL File entry for the first transport global in the distribution.
KIDS starts processing first transport global.
Answer pre-install questions (if developer needs any answers for pre-install phase of installation).
Answer Data Dictionary (DD) and Data questions (if developer is exporting DDs or data).
Answer post-install questions (if developer needs any answers for post-install phase of installation).
Any more transport globals to process?
Yes
No
Question: Disable options and protocols during install?
Yes
Edit/select set of options and protocols to disable during install. If you select anything to disable, TaskMan will also be suspended (from running scheduled options only), during install.
No
Question: Move routines to other CPUs?
Yes
Select CPUs to install routines on.
No
Queue job
Installation task created, at time you specify.
Question: Device for Installation?
Up-arrow out
Abort Installation.
Question: Remove transport global(s) from ^XTMP?
No
Run job directly
Installation runs immediately, using the current device.
Yes
KIDS removes transport global(s) from ^XTMP
Exit

24.7.14 Installation Progress

If the device selected for output is a VT100-compatible (or higher) terminal, KIDS displays the installation output in a virtual window on the terminal. Below the virtual window, a progress bar graphically illustrates the percentage complete that the current part of the installation has reached. KIDS is able to report progress for the installation of files and for all components (PRINT templates, forms, help frames, routines, options, etc.) KIDS lists those compiled cross-references, INPUT templates, and PRINT templates that were created during the install process. KIDS does *not* show progress for installing data, nor for pre- and post-install tasks.

On all other devices, progress is reported using dots.

Figure 324: Installation Progress—Sample Output



```
                           TEST 1.1
Installing Routines:
               Oct 07, 2004@15:00:02

 Installing PACKAGE COMPONENTS:

 Installing PRINT TEMPLATE
               Oct 07, 2004@15:00:04

 Updating Routine file...

 The following Routines were created during this install:
     ZZR4

 Updating KIDS files...

 TEST 1.1 Installed.
               Oct 07, 2004@15:00:05

 100%
Complete         25          50          75
```

24.7.15 Once the Installation Finishes

When the installation runs, its output is sent to the device you specified when you answered the installation questions. If, for example, you queued the installation to a printer, the output is sent to the printer.

You can find out whether an installation finished by looking up the entry in the INSTALL (#9.7) file for that installation (use the **Install File Print** [XPD PRINT] option). You should check whether an installation completed successfully or not:

- If the install completed successfully, the STATUS (#.02) field in the INSTALL (#9.7) file entry is set to "**Install Completed**."
- If the install errored out, the STATUS (#.02) field in the INSTALL (#9.7) file entry is still set to "**Install Started**." If it errored out, you need to find out what went wrong and restart the installation.

REF: For information on restarting an installation, see the "Restarting Aborted Installation" section.

If you disabled scheduled options, options, and protocols, KIDS should have re-enabled those (unless the install errored out).

You should refer to the instructions that came with the software you installed to see what post-installation tasks, if any, you should perform.

24.7.16 Restarting Aborted Installations

A feature of KIDS is the ability to restart an aborted installation. KIDS uses a checkpoint system to keep track of how many phases of an installation it completed. When an installation aborts for some reason, you can restart the installation (using the **Restart Install of Package(s)** [XPD RESTART INSTALL] option). KIDS does *not* automatically re-run the entire installation from the beginning; instead, it re-runs the installation only from the last completed checkpoint.

As well as some standard checkpoints built into KIDS (e.g., completion of pre-install, completion of each component type, and completion of post-install), KIDS lets developers create checkpoints for use within their pre- and post-install routines. So depending on how the developer has designed a pre- or post-install, it is possible that, when re-started, the pre- or post-install does *not* have to be re-run in its entirety either (if the error occurred there). Instead, KIDS only re-runs the pre- or post-install from the last completed developer checkpoint (if any) within the pre- or post-install.

Before restarting an installation, you should try to determine what caused the installation to abort. If an error occurred, any error messages are in the INSTALL (#9.7) file entry, in the MESSAGES word-processing-type field. Once you've fixed the problem, you can use the **Restart Install Of Package(s)** [XPD RESTART INSTALL] option to continue with the installation. KIDS also allows you to restart an install if the install is queued and you get an error during the installation.

24.7.17 Recovering from an Aborted Distribution Load

If you encounter an error while loading a distribution (using KIDS' Load a Distribution [XPD LOAD DISTRIBUTION] option from the export medium into the **^XTMP** global), you are unable to re-load the distribution until you clear out what was stored during the aborted load attempt.

To clear out the previously loaded distribution, use the **Unload a Distribution** [XPD UNLOAD DISTRIBUTION] option. To unload a distribution, enter the name of the *first* transport global that was loaded when you loaded the distribution. The entries in the INSTALL (#9.7) file for all transport globals in the distribution are removed, and the transport globals themselves are purged from the **^XTMP** global.

Once you delete entries in the INSTALL (#9.7) file and entries in the **^XTMP** global with the **Unload a Distribution** [XPD UNLOAD DISTRIBUTION] option, you should be able to reload the distribution in question. If the install was already started and you choose to unload the distribution, you first *must* edit the INSTALL (#9.7) file and set the STATUS (#.02) field to **Loaded From Distribution** (i.e., **0**) prior to using the **Unload a Distribution** [XPD UNLOAD DISTRIBUTION] option.

24.8 Installing Global Distributions

The second type of distribution supported by KIDS is called a global distribution. This type of distribution, unlike standard distributions, is used to only export globals.

You still use the **Load a Distribution** [XPD LOAD DISTRIBUTION] option to install global distributions. Unlike loading a standard distribution, however, KIDS installs global distributions immediately from the **Load a Distribution** [XPD LOAD DISTRIBUTION] option. Also, there is no queueing of the installation.

A global distribution can only contain one transport global, and the transport global can only export globals. You know that the distribution you're installing is a global distribution rather than a standard distribution, because when you load it with the **Load a Distribution** [XPD LOAD DISTRIBUTION] option, KIDS displays the message shown in Figure 325:

Figure 325: KIDS Global Distribution—Sample Message

```
This is a Global Distribution. It contains Global(s) that will
update your system at this time. The following Global(s) will be installed:
```

The **Load a Distribution** [XPD LOAD DISTRIBUTION] option lists each global that will be installed from the distribution. Each global in the list is marked **OVERWRITE** or **REPLACE**:

- **OVERWRITE**—Load the global *without* purging the site's version of the global beforehand.
- **REPLACE**—Purge the site's version of the global first, and then load the global.

You are given two chances to abort the installation of the global distribution. If you answer **YES** to both questions, the globals in the global distribution are installed immediately.

24.9 Purging the BUILD and INSTALL Files

Each KIDS installation adds one entry to the BUILD (#9.6) and INSTALL (#9.7) files for every transport global installed from the distribution.

REF: For information about purging these files, see the discussion of the **Purge Build or Install Files** option in the "Purge Build or Install Files" section in the "KIDS: System Management—Utilities" section.

Figure 326: Installation of a Global Distribution—Load a Distribution Option

```
Select Installation Option: LOAD A DISTRIBUTION
Enter a Host File: [DMANAGER]XGGLOBAL.DAT

KIDS Distribution save on Jan 26, 2004@12:58:25
Comment: GLOBAL PACKAGE

This Distribution contains the following Transport global(s):
     GLOBAL PACKAGE 1.0

This is a Global Distribution. It contains Global(s) that will
update your system at this time. The following Global(s) will be installed:
^XGRON(1)      Overwrite
^XGRON("PX")   Replace
^XGRON("TX")   Overwrite

If you continue with the Load, the Global(s) will be
Installed at this time.

Want to Continue with Load? YES// <Enter>
Loading Distribution...


Globals will now be installed, OK? YES// <Enter>


Installing Globals...
               Jan 26, 2004@13:04:16


 GLOBAL PACKAGE 1.0 Installed.
               Jan 26, 2004@13:04:17


Select Installation Option:
```

24.10 Alpha/Beta Tracking

Kernel provides a mechanism for tracking and monitoring installation and option usage during the alpha and beta testing phases of VistA software applications. This tool is primarily intended for application developers to use in monitoring the testing process at local test sites.

NOTE: In VA terminology "Alpha" and "Beta" testing are defined as follows:

- **Alpha Testing—**VistA test software application is running in a site's Test account.
- **Beta Testing—**VistA test software application is running in a site's Production account.

Alpha/Beta Tracking provides the following services to both developers and system administrators:

- Notification when a new alpha or beta software version is installed at a site.
- Periodic option usage reports for alpha or beta options being tracked.
- Periodic listings of errors in the software's namespace that are currently in alpha or beta test at the site.

The following options are provided on the **Alpha/Beta Test Option Usage Menu** [XQAB MENU], which is located on the **Operations Management** [XUSITEMGR] menu. These options allow developers and system administrators to monitor Alpha/Beta Tracking at a site:

- **Errors Logged in Alpha/Beta Test (QUEUED)** [XQAB ERROR LOG XMIT] option
- **Actual Usage of Alpha/Beta Test Options** [XQAB ACTUAL OPTION USAGE] option
- **Low Usage of Alpha/Beta Test Options** [XQAB LIST LOW USAGE OPTS] option
- **Print Alpha/Beta Errors (Date/Site/Num/Rou/Err)** [XQAB ERR DATE/SITE/NUM/ROU/ERR] option
- **Send Alpha/Beta Usage to Programmers** [XQAB AUTO SEND] option

REF: For more detailed information about and description of the Alpha/Beta Tracking functionality (e.g., starting, stopping, and monitoring options), see the "Alpha/Beta Tracking" section in the "KIDS: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*.

25 KIDS: System Management—Utilities

KIDS provides the following **Utilities** [XPD UTILITY] menu options shown in Figure 327:

Figure 327: KIDS Utilities Menu Options

```
Kernel Installation and Distribution System...                          [XPD MAIN]
  Utilities...                                                        [XPD UTILITY]
    Build File Print                                              [XPD PRINT BUILD]
    Install File Print                                     [XPD PRINT INSTALL FILE]
    Edit Install Status                                          [XPD EDIT INSTALL]
    Convert Loaded Package for Redistribution                 [XPD CONVERT PACKAGE]
    Display Patches for a Package                       [XPD PRINT PACKAGE PATCHES]
    Purge Build or Install Files                                   [XPD PURGE FILE]
    Rollup Patches into a Build                                [XPD ROLLUP PATCHES]
    Update Routine File                                        [XPD ROUTINE UPDATE]
    Verify a Build                                               [XPD VERIFY BUILD]
    Verify Package Integrity                                 [XPD VERIFY INTEGRITY]
```

These utilities can be used both by developers and by sites who install software created by KIDS.

25.1 Build File Print Option

The **Build File Print** [XPD PRINT BUILD] option prints out the build entry for a software application. It lists the complete definition of the software, including all files, components, install questions, and the environment pre-install and post-install routines, as shown in Figure 328.

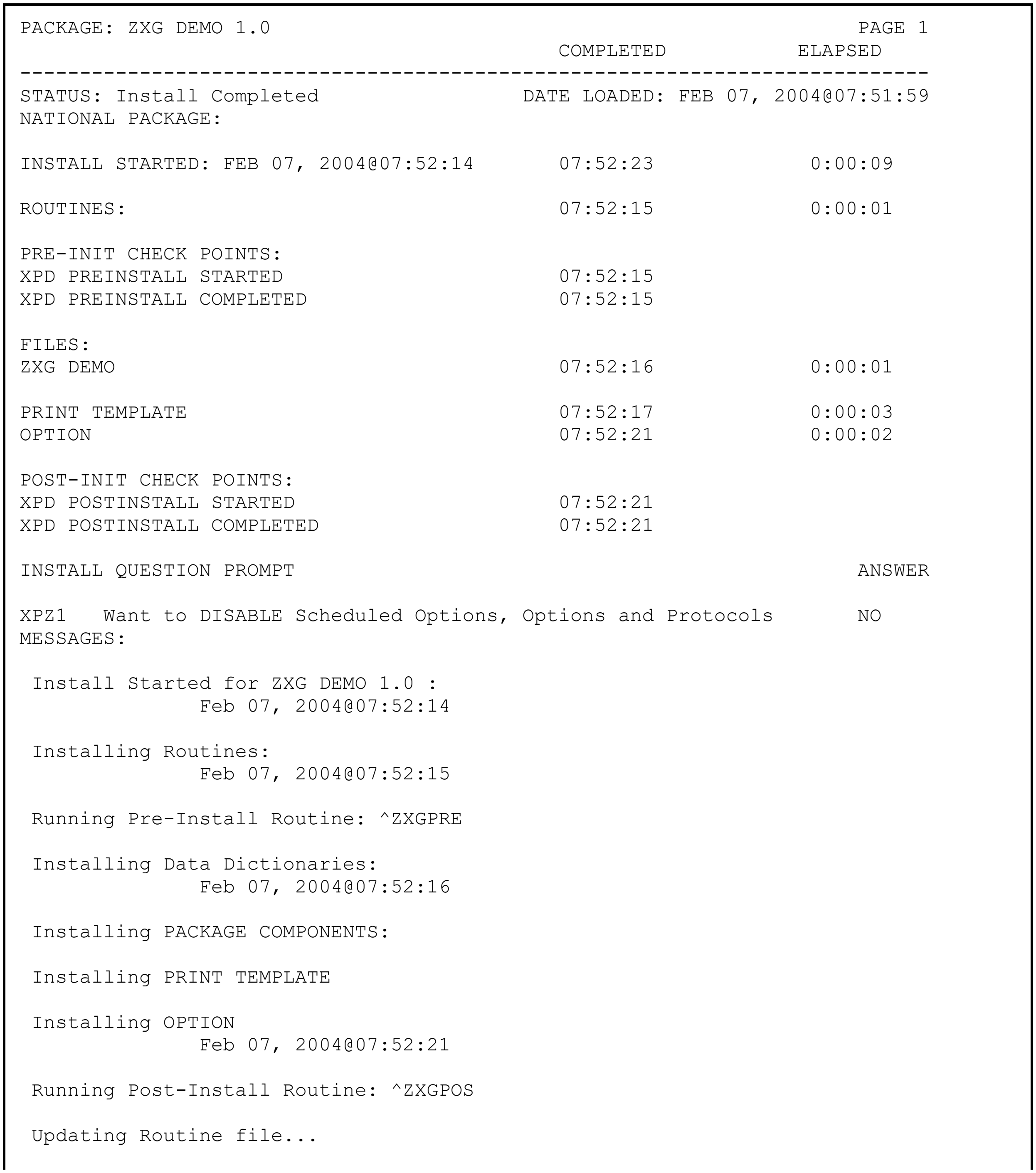
Figure 328: Build File Print Option—Sample Output

```
PACKAGE: ZXG DEMO 1.0                                                 PAGE 1
----------------------------------------------------------------------------
NATIONAL PACKAGE:
DESCRIPTION:
Package containing demonstration of ZXG* functions.

ENVIRONMENT CHECK : ZXGENV
PRE-INIT ROUTINE : ZXGPRE
POST-INIT ROUTINE: ZXGPOS


                                        UP    SEND  DATA                USER
                                        DATE  SEC.  COMES  SITE   RSLV  OVER
FILE #      NAME                        DD    CODE  W/FILE DATA   PTS   RIDE
----------------------------------------------------------------------------

662105      ZXG DEMO                    YES   YES   NO


PRINT TEMPLATE:
   ZXG PRINT    FILE #662105                     SEND TO SITE

ROUTINE:
   ZXGC00                                        SEND TO SITE
   ZXGC01                                        SEND TO SITE
   ZXGC02                                        SEND TO SITE
   ZXGC03                                        SEND TO SITE
   ZXGC04                                        SEND TO SITE
   ZXGC05                                        SEND TO SITE
   ZXGC06                                        SEND TO SITE
   ZXGC07                                        SEND TO SITE
   ZXGC08                                        SEND TO SITE

OPTION:
   ZXG TEST                                      SEND TO SITE

INSTALL QUESTIONS:
```

25.2 Install File Print Option

The **Install File Print** [XPD PRINT INSTALL FILE] option prints out the results of an installation, as stored in the INSTALL (#9.7) file. Use this option to check on the status of an installation in progress or to print out the results of a completed installation, as shown in Figure 329.

Figure 329: Install File Print Option—Sample Output



```
PACKAGE: ZXG DEMO 1.0                                                 PAGE 1
                                             COMPLETED           ELAPSED
----------------------------------------------------------------------------
STATUS: Install Completed                 DATE LOADED: FEB 07, 2004@07:51:59
NATIONAL PACKAGE:

INSTALL STARTED: FEB 07, 2004@07:52:14       07:52:23             0:00:09

ROUTINES:                                    07:52:15             0:00:01

PRE-INIT CHECK POINTS:
XPD PREINSTALL STARTED                       07:52:15
XPD PREINSTALL COMPLETED                     07:52:15

FILES:
ZXG DEMO                                     07:52:16             0:00:01

PRINT TEMPLATE                               07:52:17             0:00:03
OPTION                                       07:52:21             0:00:02

POST-INIT CHECK POINTS:
XPD POSTINSTALL STARTED                      07:52:21
XPD POSTINSTALL COMPLETED                    07:52:21

INSTALL QUESTION PROMPT                                               ANSWER

XPZ1   Want to DISABLE Scheduled Options, Options and Protocols       NO
MESSAGES:

 Install Started for ZXG DEMO 1.0 :
               Feb 07, 2004@07:52:14

 Installing Routines:
               Feb 07, 2004@07:52:15

 Running Pre-Install Routine: ^ZXGPRE

 Installing Data Dictionaries:
               Feb 07, 2004@07:52:16

 Installing PACKAGE COMPONENTS:

 Installing PRINT TEMPLATE

 Installing OPTION
               Feb 07, 2004@07:52:21

 Running Post-Install Routine: ^ZXGPOS

 Updating Routine file...
 Updating KIDS files...

 ZXG DEMO 1.0 Installed.
               Feb 07, 2004@07:52:23
```

25.3 Edit Install Status Option

The **Edit Install Status** [XPD EDIT INSTALL] option, released with Kernel patch XU*8.0*539, lets you edit the STATUS (#.02) and the INSTALL COMPLETE TIME (#17) fields in the INSTALL (#9.7) file. Use this option to change the status of a patch that was de-installed, as shown in Figure 330.

Figure 330: Edit Install Status Option—Sample User Dialog

```
Select Utilities Option: EDIT INSTALL <Enter> Status

Select INSTALL NAME: USER TEST
     1   USER TEST 1.0       Install Completed    5/14/08@11:21:04
     => TEST  ;Created on May 14, 2008@11:03:58
     2   USER TEST 1.0       Loaded from Distribution    7/8/09@10:33:16
     => TEST  ;Created on Jul 08, 2009@10:31:50
CHOOSE 1-2: 1 <Enter> USER TEST 1.0     Install Completed    5/14/08@11:21:04
     => TEST  ;Created on May 14, 2008@11:03:58
STATUS: Install Completed// ???
        This is the status of this package at this site.

     Choose from:
       0        Loaded from Distribution
       1        Queued for Install
       2        Start of Install
       3        Install Completed
       4        De-Installed
STATUS: Install Completed// <Enter>
INSTALL COMPLETE TIME: MAY 14,2008@11:21:04//
```

25.4 Convert Loaded Package for Redistribution Option

Use the **Convert Loaded Package for Redistribution** [XPD CONVERT PACKAGE] option to add software to an existing distribution.

A KIDS distribution can transport one or more software applications. What if you want to add additional software to an existing distribution? For example, suppose you have a distribution for a software application. Further, suppose that patches are transported as individual KIDS software, and you want to add all existing patches to the software's distribution? The **Convert Loaded Package for Redistribution** [XPD CONVERT PACKAGE] option lets you do this.

In Figure 327 and Figure 328, distributions for a software application (i.e., ZXG 1.0) and a patch (i.e., ZXG*1.0*1) are both loaded. The **Convert Loaded Package for Redistribution** [XPD CONVERT PACKAGE] option is used to build a new distribution combining both original distributions.

Follow these steps to create a new distribution from existing distributions:

1. Load the original distributions (there is no need to install them, however).

   In this example, you would load the distributions for ZXG 1.0 and ZXG*1.0*1 (but you would *not* install them).

2. Use the **Convert Loaded Package for Redistribution** [XPD CONVERT PACKAGE] option. It lets you choose loaded transport globals and transfers them into a format ready for export. Also, it creates build entries for each software application contained in the distributions. This allows you to create a new distribution containing the transport globals from the existing distributions. Kernel patch XU*8.0*44 added the “Want to make the Transport Globals Permanent? NO//” prompt, answering **YES** to this prompt flags the global so that it is *not* deleted after the transportation. This provides a “**Gold**” account or library of software and patches that are included in a Transport Global.

   In the example in Figure 331, you would first convert the loaded distribution **ZXG 1.0** into a form ready to re-distribute:

Figure 331: Convert Loaded Package for Redistribution—Sample User Dialog (1 of 2)

```
Select Utilities Option: CONVERT LOADED PACKAGE FOR REDISTRIBUTION
Select INSTALL NAME: ZXG 1.0 <Enter>         Loaded from Distribution

This distribution was loaded on Feb 28,2004@08:15:05 with header of

It consisted of the following Install(s):
ZXG 1.0

Want to make the Transport Globals Permanent? NO// YES
Want to continue with the conversion of the package(s)? NO// YES
  ** DONE **

Select Utilities Option:
```

Then you would convert the patch distribution, **ZXG*1.0*1**, into a form ready to re-distribute:

Figure 332: Convert Loaded Package for Redistribution—Sample User Dialog (2 of 2)

```
Select Utilities Option: CONVERT LOADED PACKAGE FOR REDISTRIBUTION
Select INSTALL NAME: ZXG*1.0*1 <Enter>         Loaded from Distribution

This distribution was loaded on Feb 28,2004@08:15:35 with header of

It consisted of the following Install(s):
ZXG*1.0*1

Want to make the Transport Globals Permanent? NO// YES
Want to continue with the conversion of the package(s)? NO// YES
  ** DONE **
```

3. Create the new distribution with the **Transport a Distribution** [XPD TRANSPORT PACKAGE] option. Select each build from the original distributions that you want to be part of the new distribution. For each build that you select, you should be told that the transport global already exists and be asked if you want to use this transport global. Answer **YES** in each case to use the current transport global.

   Once you have selected all of the builds for the new distribution, go ahead and create the new distribution.

   In the example in Figure 333, you create a new distribution containing both **ZXG 1.0** (the original software application) and **ZXG*1.0*1** (an added software application):

Figure 333: Transport a Distribution—Sample User Dialog

```
Select Edits and Distribution Option: TRANSPORT A DISTRIBUTION

Enter the Package Names to be transported. The order in which they are
entered will be the order in which they are installed.

First Package Name: ZXG 1.0 <Enter> **Transport Global exists**
    Use this Transport Global? YES
Another Package Name: ZXG*1.0*1 <Enter> **Transport Global exists**
    Use this Transport Global? YES
Another Package Name: <Enter>

Order
  1    ZXG 1.0    **will use current Transport Global**
  2.   ZXG*1.0*1    **will use current Transport Global**

OK to continue? NO//YES

Enter a Host File: ZXG1.KID
Header Comment: PATCHED DISTRIBUTION ZXG 1.0

     ZXG 1.0...
     ZXG*1.0*1...

Package Transported Successfully
```

NOTE: Changing a distribution’s build entries before redistributing is *not* recommended.

25.5 Display Patches for a Package Option

The **Display Patches for a Package** [XPD PRINT PACKAGE PATCHES] option prints all patches installed for a software application. It displays the date installed and who installed the patches. It optionally prints the description of the patch. All the displayed information comes from the PACKAGE (#9.4) file, as shown in Figure 334.

Figure 334: Display Patches for a Package Option—Sample User Dialog

```
Select Utilities Option: DISPLAY PATCHES FOR A PACKAGE
Select PACKAGE NAME: KERNEL
Select VERSION: 8.0// <Enter>          07-29-95
Do you want to see the Descriptions? NO// <Enter>
DEVICE: HOME// <Enter> SYSTEM


PACKAGE: KERNEL     Oct 09, 2004 1:32 pm                            PAGE 1
PATCH #             INSTALLED                     INSTALLED BY
-----------------------------------------------------------------
VERSION: 8.0        JUL 29, 2004                  XUUSER,TEN

   28               APR 25, 2004                  XUUSER,NINE
   20 SEQ #23       FEB 09, 2004                  XUUSER,NINE
   32 SEQ #24       MAY 15, 2004                  XUUSER,NINE
   23 SEQ #25       MAY 17, 2004                  XUUSER,TEN
   39 SEQ #26       JUL 19, 2004                  XUUSER,ELEVEN
   26 SEQ #27       JUN 01, 2004                  XUUSER,TEN
   27 SEQ #28       JUN 13, 2004                  XUUSER,NINE
   24 SEQ #29       JUN 30, 2004                  XUUSER,TEN
   40 SEQ #30       AUG 28, 2004                  XUUSER,ELEVEN
   41 SEQ #31       AUG 29, 2004                  XUUSER,TEN
   29 SEQ #32       AUG 30, 2004                  XUUSER,NINE
```

25.6 Purge Build or Install Files Option

Each KIDS installation adds one entry to the BUILD (#9.6) and INSTALL (#9.7) files for every transport global installed from the distribution. You can use the **Purge Build or Install Files** [XPD PURGE FILE] option to purge entries in these files.

The first question the option asks is which file to purge, the BUILD (#9.6) or INSTALL (#9.7) file. Choose one of these files.

The next question asked is the number of versions to retain.

25.6.1 Versions to Retain

When you choose to retain some number entries for a software application, the option *must* decide which entries are most recent. The **Purge Build or Install Files** [XPD PURGE FILE] option uses numeric order based on software version number to decide which entries are the most recent. When there are multiple entries for the same version number (e.g., alpha or beta installs took place), the following order of precedence is used:

1. Released Version is the most recent (version number contains no letters, such as 8.0)
2. Beta Test Version (version number contains **V**, such as **8.0V10**)

3. Alpha Test Version (version number contains **T**, such as **8.0T10**)

25.6.2 Selecting Software Names for Purging

After versions to retain, the next prompt is “Package Name.” You can enter a partial or full software application name. You continue to be prompted for additional software names until you simply press the **<Enter>** key without making any further entries at the “Package Name” prompt.

- **Packages (Software)**—To select software entries for purging, at the “Package Name” prompt, enter a partial or full software application name. You can optionally enter partial or full version numbers. The list of candidates for purging contains all entries (excluding patch entries) whose first characters match all characters in the software name that you specify. If you enter “**ALL**”, all software (but *not* patches) are selected for purging.
- **Patches**—Patches are a special case. To select patch entries for purging, you *must* enter the full namespace of the patch, the full version number, and an asterisk. You can optionally add a partial or full patch number after the asterisk. The list of candidates for purging contain all entries whose first characters match all characters in the string you specify.

Figure 335: Purge or Install Files Option—Sample User Dialog

```
Select Utilities Option: PURGE <Enter> Build or Install Files

      Select one of the following:

           B         Build
           I         Install
           ALL       Build & Install

Purge from what file(s): B
Versions to Retain:  (0-100): 1// 0
Package Name: ALL// ZXG
Another Package Name: <Enter> ...

Package(s) in Build file, Don't retain any versions          Page 1
--------------------------------------------------------------------
ZXG 1.0
ZXG 2.0
ZXG 3.0

OK to DELETE these entries? NO// YES

Select Utilities Option:
```

25.6.3 Purging Selected Entries

Based on the software name you enter and the number of entries you ask to retain, the **Purge Build or Install Files** [XPD PURGE FILE] option lists the software it finds to purge. If you answer **YES** to the “OK to DELETE these entries? NO//” prompt, the option purges the listed entries.

25.6.4 Reasons to Retain BUILD and INSTALL File Entries

- **BUILD File**—Entries in the BUILD (#9.6) file are created by the software developers and identify every component in the software. BUILD (#9.6) file entries also contain the checksums for a software application's components. You may want to retain the build entry for the most recent versions of installed software, so that you can verify the checksums of the loaded software against its original checksums.
- **INSTALL File**—Each entry in the INSTALL (#9.7) file contains a record of the installation for a given software application. This information is useful as a record of each installation.

25.7 Rollup Patches into a Build Option

The **Rollup Patches into a Build** [XPD ROLLUP PATCHES] option finds all the patches for a software application and add their individual BUILD (#9.6) file definitions to the software's BUILD (#9.6) file definition. This enables you to create a single BUILD (#9.6) file entry that contains the definition for the patched software.

KIDS checks the BUILD (#9.6) file and lists all KIDS patches with a matching software name and version number. The list of patches is *not* necessarily displayed in patch sequence number.

This list only includes KIDS patches. Also, it does *not* include any pre- or post-install routines. You can use the **Edit a Build** [XPD EDIT BUILD] option to further modify the build and add any additional patches.

Figure 336: Rollup Patches into a Build Option—Sample User Dialog

```
Select Utilities Option: ROLLUP PATCHES INTO A BUILD

Rollup patches into Build: KERNEL 8.0T20 <Enter>     KERNEL
This package already contains the following patches:
   XU*8.0T20*4

The following patches can be rolled into Package RON 8.0T20
     XU*8.0T20*5
     XU*8.0T20*6
     XU*8.0T20*7
     XU*8.0T20*8
     XU*8.0T20*11

OK to continue? YES// <Enter>
...SORRY, HOLD ON...............................................................
................Done.
```

25.8 Update Routine File Option

The **Update Routine File** option [XPD ROUTINE UPDATE] option updates the ROUTINE (#9.8) file to match the routine set stored on the current system.

Ideally, the ROUTINE (#9.8) file would contain an entry for every routine on the current system. However, the ROUTINE (#9.8) file does *not* get updated automatically when routines are added to or deleted from the system. But KIDS needs the ROUTINE (#9.8) file so that it can store the list of routines in a software application as pointers to the ROUTINE (#9.8) file (rather than relying on namespace alone).

Developers should use this option to update the ROUTINE (#9.8) file before editing the routine component in a build entry, to ensure that all the routines they want to include in a software application can be selected by the routines' matching entries in the ROUTINE (#9.8) file.

If you answer **YES** to the question "Want me to clean up the Routine file before updating?", the option goes through the ROUTINE (#9.8) file and deletes any entries across all namespaces that have no matches with an actual routine on the current system. As of Kernel patch XU*8.0*393, however, any routine that has been marked in the CHECKSUM REPORT (#6) field in the ROUTINE (#9.8) file as "National" is *not* deleted during the clean up the Routine File phase of the update.

Then, the **Update Routine File** option [XPD ROUTINE UPDATE] option re-populates the ROUTINE (#9.8) file with all routines currently on the system for the namespaces you enter (you can exclude parts of a namespace if you want, as well).

Figure 337: Update Routine File Option—Sample User Dialog

```
Select Utilities Option: UPDATE ROUTINE FILE


Routine Namespace: XU
Routine Namespace: -XUI
Routine Namespace: <Enter>

NAMESPACE  INCLUDE                 EXCLUDE
           -------                 -------
           XU                      XUI

OK to continue? YES// <Enter>

Want me to clean up the Routine File before updating? YES// <Enter>
...SORRY, THIS MAY TAKE A FEW MOMENTS...    ...Done.
```

25.9 Verify a Build Option

The **Verify a Build** [XPD VERIFY BUILD] option checks whether a build entry's listed components actually exist on the current system. This is useful for developers who are preparing to create a transport global. They can check that there are actual components on the system matching the components requested in the build entry, in advance of trying to create a transport global. Therefore, developers should use the **Verify a Build** [XPD VERIFY BUILD] option *before* creating transport globals from build entries.

For any component in the build entry that does *not* actually exist on the system, the option outputs a one-line message identifying the missing component, with the appellation ****NOT FOUND****. The developer is also prompted with "Do you want to remove the missing Files? NO//". This allows you to verify if the missing component should in fact be removed from the build. If the missing component is required, the developer should create the missing component for the build entry before creating a transport global.

Figure 338: Verify a Build Option—Sample User Dialog

```
Select Utilities Option: VERIFY A BUILD
Select BUILD NAME: XU*8.0*11 <Enter>     KERNEL
 File #8995  ** NOT FOUND **
Do you want to remove the missing Files? NO// <Enter>

  ** DONE **

Select Utilities Option:
```

25.10 Verify Package Integrity Option

You can use the **Verify Package Integrity** [XPD VERIFY INTEGRITY] option to compare checksums of software components on the system against the checksums of the components when they were originally transported. Any discrepancies are reported. Currently, routines are the only components that are checked, but checksums are extended to other software components in the future.

The checksums of components for the currently installed software are verified against checksums stored in the BUILD (#9.6) file entry for the software. If the most recent version of the BUILD (#9.6) file entry for a software application has been purged, the **Verify Package Integrity** [XPD VERIFY INTEGRITY] option is no longer able to verify checksums for the loaded software. Because of this, in most cases you should *not* purge the most recent build entry for a software application.

As of Kernel patch XU*8.0*369, the integrity checking CHECK1^XTSUMBLD routine supports the **Compare local/national checksums report** [XU CHECKSUM REPORT] option.

As of Kernel patch XU*8.0*393, KIDS was modified to send a message to a server on FORUM when a KIDS build is sent to a Host File Server (HFS) device. This message contains the checksums for the routines in the patch. The server on FORUM matches the message with a patch if the sending domain is authorized on FORUM. There is no longer a need for developers to manually include routine checksums (either CHECK^XTSUMBLD or

CHECK1^XTSUMBLD routines) in the patch description. The patch module includes the before and after CHECK1^XTSUMBLD values in the Routine Information section at the end of the patch document.

With changes in the National Patch Module (NPM) on FORUM, when the patch is released the checksums for the routines are moved to the ROUTINE (#9.8) file on FORUM. The checksum “before” values come from the FORUM ROUTINE (#9.8) file and are considered the GOLD standard for released checksums. The local site’s **Compare local/national checksums report** [XU CHECKSUM REPORT] option uses the FORUM ROUTINE (#9.8) file as its source to create reports showing any routines that do *not* match.

This patch also modified the KIDS BUILD (#9.6) file by adding the TRANSPORT BUILD NUMBER (#63) field used to store a build number that is incremented each time a build is made. This build number is added to the second line of each routine in the **7th** “**;**” piece. This makes it easy to tell if a site is running the current release during testing and afterword. The leading “**B**” found in the checksum tells the code what checksum API to use.

VI. Toolkit

This section provides descriptive information about the set of software utilities furnished by Kernel Version 8.0 and Kernel Toolkit Version 7.3 (a.k.a. “Toolkit”), describing how these tools can be used for the management and definition of development projects.

The major areas of the Kernel Toolkit described in this section are listed below:

- **Multi-Term Look-Up (MTLU):**

  Multi-Term Look-Up (MTLU) utilities provide a method of enhancing the lookup capabilities of associated VA FileMan files. Multi-Term Look-Up (MTLU) is an adaptation of a tool developed by the Indian Health Service (IHS), which was originally made generic by the Albany Office of Information Field Office (OIFO). MTLU does the following:

  - Tests ICD diagnosis and procedure codes, CPT codes, and other commonly used references that have been entered in the LOCAL LOOKUP (#8984.4) file. Optionally, terms or phrases can be entered into the LOCAL KEYWORD (#8984.1), LOCAL SHORTCUT (#8984.2), or LOCAL SYNONYM (#8984.3) files.
  - Prints a list of shortcuts, keywords, or synonyms from a specified reference file in the LOCAL LOOKUP (#8984.4) file.
  - Adds or deletes a reference file from a site’s LOCAL LOOKUP (#8984.4) file.
  - Enters new or edit existing shortcuts, keywords, or synonyms to the LOCAL LOOKUP (#8984.4) file.

- **Routine Tools:**

  Routine Tools provide a set of generic tools to aid the VistA development community and system administrators in analysis, writing, and testing of code. These tools are used by VistA developers to support distinct tasks. Routine Tools do the following:

  - Promote standard program interfaces.
  - Check adherence to programming standards and correct syntax with the XINDEX utility.
  - Provide standard error trapping, storing, and reporting.
  - Customize and tunes site parameters for local requirements.
  - Provide M function libraries.
  - Provide a portable routine and global editor.
  - Provide a Kermit file transfer utility.
  - Provide a Multi-Term Look-Up (MTLU) utility for enhanced VA FileMan lookups.
  - Provide software project management utilities.

- **Verification Tools:**

  Verification Tools are a set of generic tools to aid the VistA development community and system administrators in reviewing M code. These tools are used by VistA developers to support distinct tasks. Verification Tools provide the following:

  - Tools used for comparison of routines and data dictionaries.
  - A tool used to record routine text indicated in the file used to maintain changes in routines.

Where applicable, each major area of Kernel Toolkit is described first in terms of its user interface then in terms of system management implications, showing the menu that can be used to accomplish the task at hand.

REF: Kernel and Kernel Toolkit Application Program Interfaces (APIs) are documented in the "Toolkit: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet Website.

NOTE: The *Parameter Tools Supplement to Patch Description (Patch XT*7.3*26)* explains the functions available with the use of the Parameter Tools, provides information on the Kernel PARAMETERS (#8989.5) file, and describes the associated Application Program Interfaces (APIs).

REF: This documentation can be downloaded from the VA Software Document Library (VDL) at: VDL Kernel Toolkit Application Documents

The following Kernel Toolkit sections were removed from the "Toolkit" section, because they are superseded by subsequent software and documentation changes:

- **Duplicate Record Merge:**

  The Kernel Toolkit "Duplicate Record Merge" documentation is superseded by the *Duplicate Record Merge: Patient Merge* software/documentation (i.e., Kernel Toolkit patch XT*7.3*23).

  The Duplicate Record Merge functionality provides a developer Merge Shell with options that allow users to check data files for duplicate entries and merge those entries if any are found. These options provide functionality to combine duplicate records based on conditions established in customized applications. The Merge Shell was originally developed by Indian Health Service (IHS) to support their Multi-Facility Integration Project.

  REF: For instructions on how to build a merge capability for a file, see the "Developing a File Merge Capability" section in the *Kernel 8.0 and Kernel*

Toolkit 7.3 Developer's Guide available on the VA Software Document Library (VDL) at: VDL Kernel Application Documents

REF: The *Duplicate Record Merge: Patient Merge* documentation is available on the VDL at: VDL Duplicate Record Merge: Patient Merge Application Documents

- **Capacity Management:**

  The Kernel Toolkit "Capacity Management" documentation is superseded by the following software/documentation:

  - Capacity Management (CM) Tools 3.0
  - Resource Usage Monitor (RUM) 2.0
  - Statistical Analysis of Global Growth (SAGG) 2.0
  - VistA System Monitor (VSM) 2.0

  REF: The Capacity Management-related documentation is available on the VDL at:

  - Capacity Management (CM) Tools: VDL CM Tools Application Documents
  - Resource Usage Monitor (RUM): VDL RUM Application Documents
  - Statistical Analysis of Global Growth (SAGG): VDL SAGG Application Documents
  - VistA System Monitor (VSM): VDL VSM Application Documents

Kernel Toolkit Patch XT*7.3*102 removed all options, routines, and files associated with the following menus and options:

- VPM VAX/ALPHA Capacity Management
- Move Host File to Mailman [XTCM DISK2MAIL]
- Response Time Log Options

The following namespace options and routines were also removed:

- **XUCM***
- **XUCS***
- **XURTL***
- **XTCM DISK2MAIL** (option)

- **XTCMXTCMFILN** (routine)

Data dictionaries and data have been deleted for the following VA FileMan compatible files:

- Global **^XUCM**:
    - CM DAILY STATISTICS (#8986.6)
    - CM DISK DRIVE RAW DATA (#8986.5)
    - CM METRICS (#8986.4)
    - CM NODENAME RAW DATA (#8986.51)
    - CM SITE DISKDRIVES (#8986.35)
    - CM SITE NODENAMES (#8986.3)
    - CM SITE PARAMETERS (#8986.095)
    - VPM RESPONSE TIME DATA (#8986.098)

- Global **^%ZRTL**:
    - RESPONSE TIME (#3.091)
    - RT DATE_UCI,VOL (#3.092)
    - RT RAWDATA (#3.094)

Data has been deleted for the following *non*-VA FileMan compatible global:

- **^%ZRTL(3)**
- **^%ZRTL("RTH")**

NOTE: System Managers: The **^XUCM** and **%ZRTL** globals can be removed from your database after installation of Patch XT*7.3*10; however, please make sure no local routines access these globals before doing so.

26 Multi-Term Look-Up (MTLU)

26.1 Overview

This section contains an introduction and functional description, site implementation instructions for Multi-Term Look-Up (MTLU), and the option documentation.

26.2 Introduction to Multi-Term Look-Up (MTLU)

Many medical information systems depend on the standardized encoding of diagnoses and procedures for reports, searches, and statistics. The ICD DIAGNOSIS (#80), ICD OPERATION/PROCEDURE (#80.1), and CPT (#81) files are among some of the more critical files. The Multi-Term Look-Up utility increases the accessibility of the information in these files by associating user-supplied words or phrases with terms found in a more descriptive, free-text field.

Multi-Term Look-Up allows:

- Local setup of virtually any reference file.
- Modification of the behavior of the "special" lookup by defining shortcuts, synonyms, or keywords.

MTLU integrates with any software that uses a reference file that has been entered into a site's LOCAL LOOKUP (#8984.4) file.

26.3 Functional Description

The Multi-Term Look-Up (MTLU) utility provides a method of enhancing the lookup capabilities of associated software applications. This utility is comprised of the following options:

- **Multi-Term Lookup (MTLU)** [XTLKLKUP] option—Used to test ICD diagnosis and procedure codes, CPT codes, and other commonly used references that have been entered in the LOCAL LOOKUP (#8984.4) file. Optionally, terms or phrases may be entered into the LOCAL KEYWORD (#8984.1), LOCAL SHORTCUT (#8984.2) (#8984.1), or LOCAL SYNONYM (#8984.3), files.
- **Print Utility** [XTLKPRTUTL] option—Used to print a list of shortcuts, keywords, or synonyms from a specified reference file in the LOCAL LOOKUP (#8984.4) file. This list can be sorted alphabetically by name or numerically by code.
- **Delete Entries from Look-Up** [XTLKMODPARK] option—Used to delete a reference file from a site's LOCAL LOOKUP (#8984.4) file. This option should be used as a system administrator/developer utility and can only be accessed by holders of the XTLKZMGR security key.
- **Add Entries To Look-Up File** [XTLKMODPARS] option—Used to add reference files to a site's LOCAL LOOKUP (#8984.4) file. This option should be used as a system administrator/developer utility and can only be accessed by holders of the XTLKZMGR

security key. In order to add entries with this option, **DUZ(0)** *must* be set to an at-sign (**@**; Programmer access).

- **Add/Modify Utility** [XTLKMODUTL] option—Used to enter new or edit existing shortcuts, keywords, or synonyms to the LOCAL LOOKUP (#8984.4) file as described below:
    - **Shortcuts** [XTLKMODSH] option—Used to enter new or edit existing shortcuts to the LOCAL LOOKUP (#8984.4) file.
    - **Keywords** [XTLKMODKY] option—Used to enter new or edit existing keywords to the LOCAL LOOKUP (#8984.4) file.
    - **Synonyms** [XTLKMODSY] option—Used to enter new or edit existing synonyms to the LOCAL LOOKUP (#8984.4) file.

26.4 Usage Considerations

MTLU provides users and developers with the ability to perform specialized lookups on database files using standard VA FileMan calls. These files typically comprise a number or “term” in the .01 field and a longer description or definition in some other field.

In the simplest application of MTLU, a special lookup **XTLKDICL** routine is defined in the file’s data dictionary (DD), then a MUMPS cross-reference is applied to the description/definition field. Options are available to fully configure a file for use with MTLU. FileMan is used to create/build the cross-reference. To set the cross-reference, text from the selected field is passed to a tokenizing **XTLKTOKN** routine. Trivial words are filtered by an expanded Key Word In Context (KWIC), and then each remaining token is added to the cross-reference.

To request a lookup, users and developers can pass in words or phrases. Their input is similarly tokenized. However, only terms associated with *all* tokens entered are found. Input can be generalized using partial words or fewer words as well as lexical variants. For example, using the VA FileMan **Inquire to File Entries** [DIINQUIRE] option on the ICD DIAGNOSIS (#80) file one could first enter “**MALIG**”. MTLU informs the user which terms apply to the search, “**MALIG/MALIGNAN**T”, and that **447** matches are found. To be more specific, the user might enter “**MALIG LIP**” to request all malignancies associated with the lip. In this case, only **12** matches are found. The user can further screen searches by using the grave accent “Not-Sign” (`) before a word or phrase. To request all malignancies of the lip *except* those of the lower lip, one could enter “**MALIG LIP ‘LOWER**” and obtain **10** matches. Though the term “**malignancies**” may *not* exist in the lookup file, MTLU might still produce a match. When a term contains a suffix that does *not* produce a match, MTLU removes the suffix and continues the search.

REF: For more information on the **Inquire to File Entries** option, see the “Print” section in the *VA FileMan User Manual*.

Three additional files are supplied that can dramatically alter the predictable behavior described above. They are checked in the following order against the user's entry:

1. LOCAL SHORTCUT (#8984.2) file: Shortcuts are used to point to a single term. They can be a word or phrase. MTLU checks the user's entry against this file first for an exact match. If found, the lookup displays only the associated entry. A single shortcut *cannot* point to multiple terms.
2. LOCAL SYNONYM (#8984.3) file: Synonyms can be associated with many terms in a file because they can be associated with multiple "tokens" rather than a specific term. For example, **CANCER** can be defined as a synonym of "**MALIG**", "**TUMOR**", and "**LEUKEMIA**". When the user enters **CANCER**, the lookup finds *all* terms associated with the three tokens as if each had been entered separately. Compared with the example above, **CANCER** returns **534** matches. **CANCER LIP** returns the same **12** matches as **MALIG LIP**.
3. LOCAL KEYWORD (#8984.1) file: A keyword or phrase can be associated with a single term, much like a shortcut; however, it can also be associated with multiple terms, and multiple keywords can be associated with the same term.

The term SMOKER can be used as a synonym or keyword. As a keyword, one can associate it with a few *specific* diseases. As a synonym, properly selected tokens might result in a display of all smoking-related diseases.

Recall that MALIG results in 447 matches. If this were used as a shortcut to a single entry, MTLU would display only that entry and the remaining 446 would never be displayed.

These files add some control over the behavior of certain lookups. However, developers should use extreme caution when placing entries in these files to ensure that results are predictable and appropriate for both users and other VistA software developers.

The decision to populate them for a given lookup file depends on whether or *not* a commonly used word or phrase results in any matches during a lookup. If *not*, it is a candidate. The LOCAL KEYWORD (#8984.1), LOCAL SHORTCUT (#8984.2), and LOCAL SYNONYM (#8984.3) files should only be populated with common words or phrases.

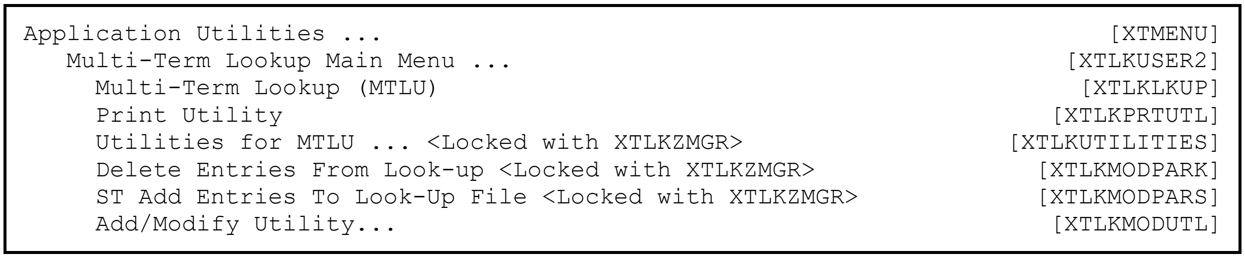
In the event that a search produces no matches, MTLU continues with a standard FileMan search.

26.5 User Interface

26.5.1 Multi-Term Look-Up Menu Options

This section describes the **Multi-Term Lookup Main Menu** [XTLKUSER2], which can be selected from the **Application Utilities** [XTMENU] menu. The options are described in the same order as they appear on the screen, as shown in Figure 339:

Figure 339: Multi-Term Lookup Main Menu Options



```
Application Utilities ...                                                  [XTMENU]
   Multi-Term Lookup Main Menu ...                                      [XTLKUSER2]
     Multi-Term Lookup (MTLU)                                            [XTLKLKUP]
     Print Utility                                                     [XTLKPRTUTL]
     Utilities for MTLU ... <Locked with XTLKZMGR>                  [XTLKUTILITIES]
     Delete Entries From Look-up <Locked with XTLKZMGR>               [XTLKMODPARK]
     ST Add Entries To Look-Up File <Locked with XTLKZMGR>            [XTLKMODPARS]
     Add/Modify Utility...                                             [XTLKMODUTL]
```

Most MTLU options are described using the following methods:

- **Introduction—**A detailed description of the option is given. The introduction usually contains any necessary special instructions.
- **Process Chart—**The step-by-step flow of the option is illustrated, showing the various choices allowed at each prompt.
- **Examples—**In most cases, there is an example of what might appear on the screen when using the particular option. If the option produces a hardcopy output, an example of the output is usually given.

The phrase "You will be prompted for a device at this step" appears in the process chart when a device is asked for. A Standard Device Chart is shown on the next page. It aids in answering prompts related to device selection.

The MTLU Process Charts do *not* contain documentation of the system's response to erroneous input. In certain instances, in order to preserve the integrity of previously entered data, the system does *not* allow the entry of a caret (^, sometimes referred to as an up-arrow). This might *not* be documented.

The chart in Figure 336 aids in answering prompts related to device selection:

26.5.1.1 Standard Device Chart

Figure 340: Standard Device Chart

```
                                      IF USER                            THEN
STEP   AT THIS PROMPT...              ANSWERS WITH...                    STEP

  1    DEVICE:                         Device name/number
                                       from your DEVICE file (#3.5)
                                       for report to print on..............3
                                       'Q'UEUE to have report
                                       queued to print at a
                                       Later date/time.....................2
                                       <Enter> for report to
                                       Print on your screen................3
                                       Up-arrow <^>........................6

  2    DEVICE:                         Device name/number from
                                       your DEVICE file (#3.5)
                                       for report to print on..............3
                                       Up-arrow <^>........................6

  3    RIGHT MARGIN: 132//            *<Enter> to accept default,
                                       different RIGHT MARGIN Value, or
                                       up-arrow <^>........................6

                    *The next step depends on what you entered in Step 1:
                                       Device name/number..................4
                                       "Q".................................5
                                       <Enter>  (The report appears on your
                                       screen).............................6

  4    WANT TO FREE UP THIS
       TERMINAL?  NO//                 <Enter> to accept default...........6
                                       'Y'ES to free up terminal
                                       during report processing
                                       and to exit from the
                                       system..............................5
                                       Up-arrow <^>........................6

  5    REQUESTED TIME TO PRINT:       *<Enter> to accept default...........6
       NOW//                          *Later date/time for report
                                       process to begin....................6
                                       Up-arrow <^>........................6

      *If <Enter> or later date/time is entered, the following
       message appears:  "REQUEST QUEUED!"

  6    Return to the menu.
```

26.5.2 Using the Multi-Term Lookup (MTLU) Option

The **Multi-Term Lookup (MTLU)** [XTLKLKUP] option is used to test the ICD diagnosis and procedure codes, CPT codes, and other commonly used references that have been entered in the LOCAL LOOKUP (#8984.4) file and have been associated with a shortcut, synonym, or keyword.

The system searches for entries in the following order:

1. Shortcut
2. Synonym
3. Keyword

If you are entering a multi-term narrative (phrase), you can enter double spaces between each term to avoid a search of the LOCAL SHORTCUT (#8984.2) file. When searching for a keyword phrase, the system searches for each word in the phrase and then displays all common entries. For example, if the keyword is FRACTURE FEMUR, the system searches for FRACTURE and then FEMUR and displays only those codes with a diagnosis containing both keywords or synonyms of those words.

The process chart in Figure 341 shows the prompts and steps involved in using the **Multi-Term Lookup (MTLU)** option:

Figure 341: Multi-Term Lookup (MTLU) Option Process Chart

```
                                      IF USER                            THEN
STEP   AT THIS PROMPT...              ANSWERS WITH...                    STEP
_____________________________________________________________________________

  1    Lookup on which file?:         Name of entry in LOCAL
                                      LOOKUP file (#8984.4)................2
                                      <?> for list of entries..............1
                                      <Enter> or up-arrow <^>..............4
_________________________________________________________________________________

  2    NARRATIVE:                     Existing shortcut,
                                      synonym, or keyword..................3

       If a word, phrase, or symbol is entered that the system cannot
       identify, the following appears:

       "Narrative contained no usable words.

       The following word(s) was not used in this search:  {word(s)}

       Search was unsuccessful."
_________________________________________________________________________________

       The selected code or description is displayed. The system searches
       in the following order: shortcut, synonym, then keyword. If more than
       one entry is found, they are displayed, and you are prompted to
       select one. If only one entry is found, the following appears:

  3    OK? Y//                        <Enter> to accept default............4
                                      'N'O.................................4
_________________________________________________________________________________

  4    Return to the menu.
```

Figure 342 is an example of what might appear on your screen when using the **Multi-Term Lookup (MTLU)** option:

Figure 342: Multi-Term Lookup (MTLU) Option—Sample User Entries

```
Lookup on which file?: ICD DIAGNOSIS

NARRATIVE: DIABETES MELLITUS
( DIABETES|DIABETIC MELLITUS )
....

The following 3 matches were found:

   1: 250.00  (250.00)
      DIABETES UNCOMPL ADULT/NIDDM
   2: 250.40  (250.40)
      DIAB RENAL MANIF ADULT/NIDDM
   3: 775.0  (775.0)
      INFANT DIABET MOTHER SYN

Select 1-3: 2
```

26.5.3 Using the Print Utility Option

The **Print Utility** [XTLKPRTUTL] option is used to print a list of shortcuts, keywords, or synonyms from a specified reference file in the LOCAL LOOKUP (#8984.4) file. Both the shortcut and keyword lists can be sorted alphabetically by name or numerically by code. The synonym list, however, only prints alphabetically.

Since these lists can be long and the generation time consuming, it is suggested you queue the report to a device during off hours.

The process chart in Figure 343 shows the prompts and steps involved in using the **Print Utility** option:

Figure 343: Print Utility Option Process Chart

```
                                      IF USER                            THEN
STEP   AT THIS PROMPT...              ANSWERS WITH...                    STEP

  1    Select one of the following:

            SH        Shortcuts
            KE        Keyword
            SY        Synonyms

        Print which file?:             SH for Shortcuts....................2
                                       KE for Keywords.....................2
                                       SY for Synonym......................3
-----------------------------------------------------------------------------
  2    Select one of the following:

            A         Alphabetic
            C         Code

        Sort By?:                      'A'lphabetic........................3
                                       'C'ode..............................3
-----------------------------------------------------------------------------
  3    Print {Shortcuts, Keywords, or
       Synonyms} for which file?:      Name of entry in LOCAL
                                       LOOKUP file (#8984.4)...............4
                                       <?> for list of entries.............3
                                       <Enter> or up-arrow <^>.............5
-----------------------------------------------------------------------------
  4    You will be prompted for a device at this step.....................1
-----------------------------------------------------------------------------
  5    Return to the menu.
```

Figure 344 is an example of what might appear on your screen when using the **Print Utility** option (an example of the output generated by this option is provided following the computer dialog in Figure 344):

Figure 344: Print Utility Option—Sample User Entries and Sample Output

```
     Select one of the following:

          SH        Shortcuts
          KE        Keywords
          SY        Synonyms

      Print which file?: SH <Enter> Shortcuts

     Select one of the following:

          A         Alphabetic
          C         Code

      Sort By?: A <Enter> lphabetic


Print Shortcuts for which file?: CPT
DEVICE:HOME// <Enter>    RIGHT MARGIN: 80// <Enter>
```

Sample output.

```
Shortcuts of the CPT file sorted by Name        NOV 23, 1994  13:36  PAGE 1
FREQUENTLY USED NARRATIVE                ENTRY
---------------------------------------------------------------------------

DREAM                                    01200
NIGHT                                    02400
SLEEP                                    01100
```

26.5.4 Using the Utilities for MTLU Option

The following is a list of the options and their descriptions that comprise the **Utilities for MTLU** [XTLKUTILITIES] menu. This menu can only be accessed by holders of the XTLKZMGR security key:

- **Delete Entries From Look-Up** [XTLKMODPARK] option—Deletes entries from the LOCAL LOOKUP (#8984.4) file. In order to do this, there *cannot* be any shortcuts, synonyms, or keywords associated with the file to be deleted. This option should be used as a system administrator/developer utility and can only be accessed by holders of the XTLKZMGR security key.
- **Add Entries To Look-Up File** [XTLKMODPARS] option—Sets entries in the LOCAL LOOKUP (#8984.4) file. This option should be used as a system administrator/developer utility and can only be accessed by holders of the XTLKZMGR security key. In order to add entries with this option, **DUZ(0)** *must* be set to an at-sign (@; Programmer access).

- **Add/Modify Utility** [XTLKMODUTL] option—Adds or edits entries in the LOCAL KEYWORD (#8984.1), LOCAL SHORTCUT (#8984.2), and LOCAL SYNONYM (#8984.3) files.

26.5.4.1 Delete Entries from Look-Up Option

The **Delete Entries From Look-Up** [XTLKMODPARK] option is used to delete a reference file from a site's LOCAL LOOKUP (#8984.4) file.

All shortcuts, synonyms, and keywords associated with the reference file you wish to delete *must* be canceled before you attempt to delete the file.

It should be noted that when a reference file is **KILL**ed through this option, all variable pointers from the LOCAL KEYWORD (#8984.1) and LOCAL SHORTCUT (#8984.2) files are deleted. The special lookup routine for the file is also deleted.

Only holders of the XTLKZMGR security key, can access this option.

NOTE: Due to the brevity of this option, no process chart has been provided.

Figure 345 is an example of what might appear on your screen when using the **Delete Entries From Look-Up** option:

Figure 345: Delete Entries From Look-Up Option—Sample User Entries

```
Select LOCAL LOOKUP NAME: PROCEDURE MODIFIERS

Are you sure you want to delete PROCEDURE MODIFIERS? YES

Deleting from Local Lookup file.....
Deleting variable pointers from Local Keyword and Shortcut files.
Deleting special lookup routine from PROCEDURE MODIFIERS DD.
```

26.5.4.2 Add Entries To Look-Up File Option

The **Add Entries To Look-Up File** [XTLKMODPARS] option is used to add/edit reference files to a site's LOCAL LOOKUP (#8984.4) file.

Examples of files that a site might wish to enter in their LOCAL LOOKUP (#8984.4) file include:

- ICD DIAGNOSIS (#80)
- ICD OPERATION/PROCEDURE (#80.1)
- CPT (#81)

Only holders of the XTLKZMGR security key, can access this option. In order to add entries with this option, **DUZ(0)** *must* be set to an at-sign (**@**; Programmer access).

The process chart in Figure 346 shows the prompts and steps involved in using the **Add Entries To Look-Up File** option:

Figure 346: Add Entries To Look-Up File Option Process Chart (1 of 2)

```
                                   IF USER                          THEN
STEP   AT THIS PROMPT...           ANSWERS WITH...                  STEP

  1    Select LOCAL LOOKUP NAME:   Name of new reference
                                   file you wish to enter
                                   in LOCAL LOOKUP (#8984.4) file......2
                                   <?> for file list...................1
                                   Name of existing file...............8
                                   <Enter> or up-arrow <^>............12

  2    ARE YOU ADDING {reference
       file name} AS A NEW LOCAL
       LOOKUP (THE nTH)?           'Y'ES...............................3
                                   'N'O................................1

  3    LOCAL LOOKUP NAME:
       {reference file name}//     <Enter> to accept default...........4
                                   Other file name.....................4

  4    LOCAL LOOKUP DISPLAY PROTOCOL: Entry point for routine
                                   to determine the display
                                   format..............................5
                                   <Enter> to accept the
                                   internal default display
                                   format..............................5

       If the entry made at this step is not the same as the
       cross reference in the description field of the file,
       the software still functions, but it only uses the
       keywords entered in the LOCAL LOOKUP (#8984.4) file.
```

*Required field

Figure 347: Add Entries To Look-Up File Option Process Chart (2 of 2)

```
                                    IF USER                            THEN
STEP   AT THIS PROMPT...            ANSWERS WITH...                    STEP


* 5    INDEX:                       Cross reference to be
                                    used to create new key-
                                    words..............................6
       NOTE:  The following message is displayed :

              "...Ok, will now setup KEYWORD and SHORTCUT file DD's to allow
              terms for {reference file name} entries..."

* 6    PREFIX: M//:                 Letter(s) to be used to
                                    identify a variable
                                    pointer............................7
---------------------------------------------------------------------------------

  7    The following reminder message is displayed:

       <REMINDER> Using 'Edit File', set the lookup routine, XTLKDICL, in
       {reference file name} DD     ...................................1

---------------------------------------------------------------------------------

       The selected file is displayed.

  8    ...OK? YES//                 <Enter> to accept default ..........9
                                    'N'O...............................1
---------------------------------------------------------------------------------

  9    LOCAL LOOKUP NAME:
      {reference file name}//       <Enter> to accept default..........10
                                    Correct file name..................10
---------------------------------------------------------------------------------

 10    LOCAL LOOKUP DISPLAY PROTOCOL:
       {protocol}//                 <Enter> to accept default..........11
                                    Correct entry point for
                                    routine to set display
                                    format.............................11
                                    <Enter> (no default) to
                                    accept the internal
                                    Default display format.............11
---------------------------------------------------------------------------------

 11    INDEX: {index}//             <Enter> to accept default..........12
                                    correct cross reference
                                    to be used to create new
                                    Keywords...........................12
---------------------------------------------------------------------------------

 12    Return to the menu.
```

*Required field

Figure 348 is an example of what might appear on your screen when using the **Add Entries To Look-Up File** option:

Figure 348: Add Entries To Look-Up File Option—Sample User Entries

```
Select LOCAL LOOKUP NAME: PROCEDURE MODIFIERS
   ARE YOU ADDING 'PROCEDURE MODIFIERS' AS A NEW LOCAL LOOKUP (THE 4th)? Y <Enter>
(YES)
   LOCAL LOOKUP NAME: PROCEDURE MODIFIERS// <Enter>
   LOCAL LOOKUP DISPLAY PROTOCOL: <Enter>
INDEX: AIHS
...Ok, will now setup KEYWORD and SHORTCUT file DD's
   to allow terms for 'PROCEDURE MODIFIERS' entries...
PREFIX: M// <Enter>
  <REMINDER> Using 'Edit File', set the lookup routine, XTLKDICL, in PROCEDURE
MODIFIERS DD
Select LOCAL LOOKUP NAME: <Enter>
```

26.5.4.3 Add/Modify Utility Option

The **Add/Modify Utility** [XTLKMODUTL] option is used to enter new or edit existing shortcuts, keywords, or synonyms to the LOCAL LOOKUP (#8984.4) file.

26.5.4.3.1 Shortcut

A **shortcut** is a word or phrase that recognizes one specific code or procedure. If you are adding a shortcut whose text duplicates the first part of an existing entry, you *must* enclose the new shortcut word or phrase in double quotes to prevent the system from matching it to existing terms.

26.5.4.3.2 Keyword

A **keyword** is a word or phrase that corresponds to several related codes or procedures. Keywords are typically terms commonly used to describe a clinical entity. Entering a series of keywords separated by single spaces results in all of the keywords being added to the specified code.

26.5.4.3.3 Synonym

A **synonym** is a word entered to expand the lookup capability of an existing term or terms in the LOCAL LOOKUP (#8984.4) file. Synonyms would be used in cases where several words within the text of codes or procedures have the same diagnostic meaning (e.g., **CANCER** and **MALIGNANCY**). A synonym can be entered for an existing keyword or for a word in the diagnostic description or procedure (e.g., the term **CANCER** might be matched to the synonyms **MALIGNANCY**, **LEUKEMIA**, and **CARCINOMA**). When **CANCER** is referenced in the **Multi-Term Lookup (MTLU)** [XTLKLKUP] option, it recognizes all the codes and descriptions associated with **MALIGNANCY**, **LEUKEMIA**, and **CARCINOMA**.

NOTE: A synonym replaces the original word in the lookup process; therefore, to retain the original word in the search, it *must* be matched to itself as well as to other synonyms.

Words used as a shortcut should never be repeated as synonyms or keywords. Since the system searches for shortcuts first and stops when one is found, it *cannot* find duplicated words in the LOCAL SYNONYM (#8984.3) or LOCAL KEYWORD (#8984.1) files. Since searching all files for each word is time consuming, the search is done in this order so as to speed up the search process.

Since the add/modify functions for Shortcuts, Keywords, and Synonyms are considered separate options, a process chart for each is provided. The charts on the following pages show the prompts and steps involved in using the options in Figure 349:

Figure 349: Add/Modify Utility Menu Options

```
Select Add/Modify Utility Option: ??

            SH Shortcuts                                                [XTLKMODSH]
            KE Keywords                                                 [XTLKMODKY]
            SY Synonyms                                                 [XTLKMODSY]
```

The **Shortcuts** [XTLKMODSH] option, one of the three selections within the **Add/Modify Utility** [XTLKMODUTL] option, is described below.

The process chart in Figure 350 shows the prompts and steps involved in using the **Add/Modify Utility** [XTLKMODUTL] option when adding or editing a shortcut:

Figure 350: Add/Modify Utility Option—Shortcuts Process Chart (1 of 2)

```
                                      IF USER                            THEN
STEP   AT THIS PROMPT...              ANSWERS WITH...                    STEP

  1        SH      Shortcuts
           KE      Keywords
           SY      Synonyms

       Select Add/Modify
       Utility Option:                SH for Shortcuts....................2
                                      <Enter> or up-arrow <^>............11

  2    Additions/Modifications to
       Shortcuts in which file?       Name of entry in local
                                      reference file......................3
                                      <?> for list of entries.............2
                                      <Enter>.............................1

  3    Select LOCAL SHORTCUT
       FREQUENTLY USED NARRATIVE:     New text you wish to use
                                      as a shortcut.......................4
                                      Existing shortcut term..............8
                                      <Enter>.............................1

  4    ARE YOU ADDING {‘text’} AS
       A NEW LOCAL SHORTCUT?          ‘Y’ES...............................5
                                      ‘N’O or <Enter>.....................3
       An at-sign (@) entered at this step deletes the entire entry.

  5    LOCAL SHORTCUT FREQUENTLY

       USED NARRATIVE: {shortcut}//   <Enter> to accept default...........6
                                      Other text..........................6

  6    LOCAL SHORTCUT ENTRY:          Name or number of entry
                                      in LOCAL LOOKUP file
                                      (#8984.4) you wish your
                                      shortcut to reference...............7
```

Figure 351: Add/Modify Utility Option—Shortcuts Process Chart (2 of 2)

```
                                     IF USER                          THEN
STEP   AT THIS PROMPT...             ANSWERS WITH...                  STEP

  7    If the selected number/name corresponds to more than one entry, they
       are shown and you are prompted to choose one. If there is only one
       corresponding entry, it is displayed and the following appears:

       “...OK? YES//                 <Enter> to accept default...........2
                                     ‘N’O................................6
--------------------------------------------------------------------------------
  8    LOCAL SHORTCUT FREQUENTLY
       USED NARRATIVE:{shortcut}//   <Enter> to accept default...........9
                                     Correct shortcut term...............9
--------------------------------------------------------------------------------
  9    LOCAL SHORTCUT ENTRY:
       {code}//                      <Enter> to accept default...........2
                                     Correct code.......................10
--------------------------------------------------------------------------------
       The selected code is displayed.

 10    ...OK? YES//                  <Enter> to accept default...........2
                                     ‘N’O................................9
--------------------------------------------------------------------------------
 11    Return to the menu.
```

The **Keywords** [XTLKMODKY] option, one of the three selections within the **Add/Modify Utility** [XTLKMODUTL] option, is described below.

The process chart in Figure 352 shows the prompts and steps involved in using the **Add/Modify Utility** [XTLKMODUTL] option when adding or editing a keyword:

Figure 352: Add/Modify Utility Option—Keywords Process Chart

```
                                IF USER                            THEN
STEP   AT THIS PROMPT...        ANSWERS WITH...                    STEP

  1        SH      Shortcuts
           KE      Keywords
           SY      Synonyms

       Select Add/Modify
       Utility Option:          KE for Keywords.....................2
                                <Enter> or up-arrow <^>.............7
------------------------------------------------------------------------
  2    Additions/Modifications to
       Keywords in which file?  Name of entry in local
                                reference file......................3
                                <?> for list of entries.............2
                                <Enter>.............................1
------------------------------------------------------------------------
  3    Which code in the {file
       name} file?              Code for which you wish
                                to enter a keyword..................4
------------------------------------------------------------------------
  4    Select LOCAL KEYWORD NAME:  New text you wish to use
                                as a keyword........................5
                                Existing keyword term...............6
                                <Enter>.............................1
------------------------------------------------------------------------
  5    ARE YOU ADDING {'text'} AS
       A NEW LOCAL KEYWORD?     'Y'ES...............................6
                                'N'O or <Enter>.....................1
------------------------------------------------------------------------
       An at-sign (@) entered at this step deletes the entire entry.

  6    LOCAL KEYWORD NAME:
       {keyword}//              <Enter> to accept default...........2
                                Correct keyword term................2
------------------------------------------------------------------------
  7    Return to the menu.
```

The **Synonyms** [XTLKMODSY] option, one of the three selections within the **Add/Modify Utility** [XTLKMODUTL] option, is described below.

The process chart in Figure 353 shows the prompts and steps involved in using the **Add/Modify Utility** [XTLKMODUTL] option when adding or editing a synonym:

Figure 353: Add/Modify Utility Option—Adding or Editing a Synonym Process Chart (1 of 2)

```
                                      IF USER                           THEN
STEP   AT THIS PROMPT...              ANSWERS WITH...                   STEP

  1        SH      Shortcuts
           KE      Keywords
           SY      Synonyms

       Select Add/Modify
       Utility Option:                SY for Synonyms......................2
                                      <Enter> or up-arrow <^>..............9
--------------------------------------------------------------------------------------
  2    Additions/Modifications to
       Synonyms in which file?        Name of entry in local
                                      reference file.......................3
                                      <?> for list of entries..............2
                                      <Enter>..............................1
--------------------------------------------------------------------------------------
       The entry made at this step must be in all upper case
       letters.

  3    Select LOCAL SYNONYM TERM:     New text you wish to use
                                      as a synonym.........................4
                                      Existing synonym term................7
                                      <Enter>..............................1
--------------------------------------------------------------------------------------
  4    ARE YOU ADDING {‘text’} AS
       A NEW LOCAL SYNONYM?           ‘Y’ES................................5
                                      ‘N’O.................................3
--------------------------------------------------------------------------------------
       An at-sign (@) entered at this step deletes the entire entry.

  5    LOCAL SYNONYM TERM:
       {synonym}//                    <Enter> to accept default............6
                                      Other text...........................6
--------------------------------------------------------------------------------------
  6    LOCAL SYNONYM
       Select SYNONYM:                Existing term in LOCAL
                                      LOOKUP file (#8984.4) for
                                      which you are entering a
                                      synonym..............................2
```

Figure 354: Add/Modify Utility Option—Adding or Editing a Synonym Process Chart (2 of 2)

```
                                   IF USER                            THEN
STEP   AT THIS PROMPT...           ANSWERS WITH...                    STEP

  7    TERM: {term entered
       at Step 3}//                <Enter> to accept default...........8
                                   Correct synonym term................8

       The entry made at this step must be in all upper case
       letters.
-----------------------------------------------------------------------------------
  8    Select SYNONYM: {term
       synonym was entered for}//  <Enter> to accept default...........2
                                   Correct term........................2
-----------------------------------------------------------------------------------
  9    Return to the menu.
```

26.5.5 Examples

The following are examples of what might appear on your screen when using the **Add/Modify Utility** [XTLKMODUTL] option:

- Example 1 shows a new shortcut entry.
- Example 2 shows a new keyword entry.
- Example 3 shows the editing of an existing synonym entry.

26.5.5.1 Example 1

Figure 355 illustrates a new Shortcut entry.

Figure 355: Shortcut Option—Sample User Entries

```
   SH     Shortcuts
   KE     Keywords
   SY     Synonyms

Select Add/Modify Utility Option: SH <Enter> Shortcuts


Additions/Modifications to Shortcuts in which file? CPT

Select LOCAL SHORTCUT FREQUENTLY USED NARRATIVE: COUGH
   ARE YOU ADDING 'COUGH' AS A NEW LOCAL SHORTCUT? Y <Enter> (YES)
   LOCAL SHORTCUT FREQUENTLY USED NARRATIVE: COUGH// <Enter>
   LOCAL SHORTCUT ENTRY: 31659

     Searching for a CPT  31659         BRONCHOSCOPIC PROCEDURES
         ...OK? YES// <Enter>  (YES)
```

26.5.5.2 Example 2

Figure 356 illustrates a new Keyword entry.

Figure 356: Keyword Option—Sample User Entries

```
   SH     Shortcuts
   KE     Keywords
   SY     Synonyms

Select Add/Modify Utility Option: KE <Enter> Keywords


Additions/Modifications to Keywords in which file?: CPT

Which code in the CPT file?: 11044 <Enter> CLEANSING TISSUE/MUSCLE/BONE
Select LOCAL KEYWORD NAME: TISSUE SKIN
   ARE YOU ADDING ‘TISSUE SKIN’ AS A NEW LOCAL KEYWORD? Y <Enter> (YES)
   LOCAL KEYWORD NAME: TISSUE SKIN// <Enter>
```

26.5.5.3 Example 3

Figure 357 illustrates editing an existing Synonym entry.

Figure 357: Synonym Option—Sample User Entries

```
   SH     Shortcuts
   KE     Keywords
   SY     Synonyms

Select Add/Modify Utility Option: SY <Enter> Synonyms


Additions/Modifications to Synonyms in which file?: CPT

Select LOCAL SYNONYM TERM: SLEEP
TERM: SLEEP// <Enter>
Select SYNONYM: DREAM// NIGHT
```

26.6 Systems Management

26.6.1 Implementation of Multi-Term Look-Up (MTLU)

This is how a user would configure a new file to be used with MTLU. The file you select would typically contain a free text field that more completely describes the record entry. Users would then use a cross-reference on this text field to perform lookups. MTLU is distinguished from FileMan in that users can enter a narrative or phrase, rather than a single term. The cross-reference can be either a VA FileMan Key Word In Context (KWIC) cross-reference, or you can create a custom MUMPS cross-reference that calls the routine, **^XTLKWIC** (shown in Figure 354). The ICD DIAGNOSIS (#80) file is used as an example.

REF: Multi-Term Look-Up (MTLU) Application Programming Interfaces (APIs) are documented in the “Toolkit: Developer Tools” section in the *Kernel 8.0 and Kernel*

Toolkit 7.3 Developer’s Guide. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet Website.

Once you are in VA FileMan, run the Cross-Reference A Field [DIXREF] option, as shown in Figure 358:

Figure 358: VA FileMan Utility Functions Option—Sample User Entries

```
Select OPTION: UTILITY FUNCTIONS
Select UTILITY OPTION: CROSS-REFERENCE A FIELD

MODIFY WHAT FILE: ICD DIAGNOSIS// <Enter> ICD DIAGNOSIS
                                          (12535 entries)
Select FIELD: DESCRIPTION

CURRENT CROSS-REFERENCE IS MUMPS ‘D’ INDEX OF FILE
CHOOSE E (EDIT)/D (DELETE)/C (CREATE): C
WANT TO CREATE A NEW CROSS-REFERENCE FOR THIS FIELD? NO// Y <Enter> (YES)
CROSS-REFERENCE NUMBER: 2// <Enter>
Select TYPE OF INDEXING: REGULAR// MUMPS
WANT CROSS-REFERENCE TO BE USED FOR LOOKUP AS WELL AS FOR SORTING? YES// N <Enter>
(NO)
SET STATEMENT: S %=“^ICD9(““AIHS”“,I,DA)” D S^XTLKWIC
KILL STATEMENT: S %=“^ICD9(““AIHS”“,I,DA)” D K^XTLKWIC
INDEX: AC// AIHS
...

NO-DELETION MESSAGE: <Enter>
DESCRIPTION: <Enter>
  Edit? NO// <Enter>

DO YOU WANT TO CROSS-REFERENCE EXISTING DATA NOW? YES// Y <Enter> (YES)
...EXCUSE ME, THIS MAY TAKE A FEW MOMENTS...
```

Figure 359: Add Entries To Look-Up File—Sample User Entries

```
>D ^XUP

Setting up programmer environment
Terminal Type set to: C-VT100

Select OPTION NAME: APP <Enter> LICATION UTILITIES  XTMENU     Application
Utilities


          Multi-Term Lookup Main Menu ...

Select Application Utilities Option: MULTI <Enter> -Term Lookup Main Menu


          Multi-Term Lookup (MTLU)
          Print Utility
          Utilities for MTLU ...

Select Multi-Term Lookup Main Menu Option: UTIL <Enter> ities for MTLU


   KL     Delete Entries From Look-up
   ST     Add Entries To Look-Up File
          Add/Modify Utility ...

Select Utilities for MTLU Option: ST <Enter> Add Entries To Look-Up File
Select LOCAL LOOKUP NAME: ICD DIAGNOSIS
  ARE YOU ADDING 'ICD DIAGNOSIS' AS A NEW LOCAL LOOKUP (THE 3RD)? Y <Enter> (YES)
  LOCAL LOOKUP NAME: ICD DIAGNOSIS// <Enter>
  LOCAL LOOKUP DISPLAY PROTOCOL: DSPLYD^XTLKKWLD
INDEX: AIHS
...Ok, will now setup KEYWORD and SHORTCUT file DD's
   to allow terms for 'ICD DIAGNOSIS' entries...
PREFIX: M// ?
     Answer must be a unique prefix, 1-10 characters in length
```

Enter the "Variable Pointer" prefix.

```
PREFIX: M// D
  <REMINDER> Using 'Edit File', set the lookup routine, XTLKDICL, in ICD
DIAGNOSIS DD
Select LOCAL LOOKUP NAME: <Enter>
```

If *all* references to a file (by all packages) are to behave as MTLU lookups, add the special lookup routine, **^XTLKDICL**, to the file's DD using the VA FileMan **Edit File** [DIEDFILE] option.

REF: For more information on the **Edit File** [DIEDFILE] option, see the "File Utilities" section in the *VA FileMan Advanced User Manual*.

Figure 360: VA FileMan Edit File Option—Sample User Entries

```
VAH,MTL>D Q^DI


VA FileMan 20.0


Select OPTION: UT <Enter> ILITY FUNCTIONS
Select UTILITY OPTION: ED <Enter> IT FILE

MODIFY WHAT FILE: ICD DIAGNOSIS// <Enter>
NAME: ICD DIAGNOSIS// <Enter>
DESCRIPTION: <Enter>
  1>Contains all valid ICD diagnosis codes.
EDIT Option: <Enter>
Select APPLICATION GROUP: <Enter>
PROGRAMMER: <Enter>
VERSION: 9// <Enter>
DATA DICTIONARY ACCESS: <Enter>
READ ACCESS: <Enter>
WRITE ACCESS: <Enter>
DELETE ACCESS: <Enter>
LAYGO ACCESS: <Enter>
AUDIT ACCESS: <Enter>
DD AUDIT? NO// <Enter>

ASK 'OK' WHEN LOOKING UP AN ENTRY? YES// <Enter> (YES)
POST-SELECTION ACTION: <Enter>
LOOK-UP PROGRAM: XTLKDICL
CROSS-REFERENCE ROUTINE: <Enter>


Select UTILITY OPTION: <Enter>
```

NOTE: The developer might elect to use MTLU only in selected instances. This is accomplished by *not* adding the special lookup routine to the file's DD. After the file has been added to the LOCAL LOOKUP (#8984.4) file, you can make a developer call to LKUP^XTLKMGR.

REF: Multi-Term Look-Up (MTLU) Application Programming Interfaces (APIs) are documented in the "Toolkit: Developer Tools" section in the *Kernel 8.0 and Kernel Toolkit 7.3 Developer's Guide*. Kernel and Kernel Toolkit APIs are also available in HTML format at a VA Intranet Website.

27 Parameter Tools

27.1 Introduction

This section describes the Parameter Tools released with Kernel Toolkit Patch XT*7.3*26. It explains the functions available with the use of the Parameter Tools, as well as providing additional explanatory material and a generic example to illustrate the use of the Parameter Tools.

Parameter Tools was designed as a method of managing the definition, assignment, and retrieval of parameters for VistA software applications. A parameter can be defined for various levels at which you want to allow the parameter described (e.g., software level, system level, division level, location level, user level).

REF: For a list and description of the Parameter Tools (XPAR) application programming interfaces (APIs), see the “Toolkit—Parameter Tools” section in the *Kernel and Kernel toolkit Developer’s Guide*.

27.2 Background

VistA software applications are designed to be used in a variety of ways. Many aspects of site activity vary from one site to another, and thus, there are many possible ways software applications can be used that also vary from one institution to another. Each site has its own requirements—its own settings for each software application. System managers *must* modify the software parameters to fit their requirements.

Previously, each software application had its own files and options, but no two software applications had the site parameters set up the same way or found in the same place. Thus, when a new software application was released, each site would have to look for the location where the settings were stored for that software. Next, they would have to look to see what settings were available and how to set them. Very little about the parameters was uniform from software to software.

With the Computerized Patient Record System (CPRS) software, the idea was born that a parameter file could be created to export with the software. The CPRS parameter file and parameter utility were subsequently modified to create a generic method of exporting and installing other VistA software applications. Most developers were willing to abandon previous methods and use this tool for software they were developing.

Whenever you have an entity with many attributes that apply to it, you can do either of the following:

- Make one big relation to represent that entity.
- Create a "binary" relation to represent the entity. The relation consists of two columns (thus the term binary), one representing the attribute and the other representing the value for that attribute. So, each tuple (i.e., a data type/data object containing two or more components) of the relation represents a single attribute and its associated value.

NOTE: This works only when the individual attributes are independent observations (have no dependencies on anything other than the key that identifies the entity). Such a relation tends to look a lot like a Windows **INI** file.

Most of the VistA parameter files were very long lists of independent values that pertained to a single entity. In most cases, this entity was the site or system on which the software was running [similar to an **INI** file]. In other cases, however, the parameter files had multiples that made things more complex. These multiples generally allow parameters to be defined at levels more specific than the site (e.g., by divisions or hospital location). It seems best to accommodate this by using both an entity identifier and parameter together to name any given value. This yields a relation with a compound key:

Entity | Parameter = Value

Finally, it seems that multiple-valued parameters (e.g., collection times) occur often enough that it is worthwhile to add a field to identify the parameter instance. So the relation becomes:

Entity | Parameter | Instance = Value

This is the relation that the PARAMETERS (#8989.5) file is intended to represent.

Software parameter files frequently maintain parameters that apply to the site, a division, or a location. In addition, many parameters that apply to individual users are kept in the NEW PERSON (#200) file. Also, many parameter values are hard-coded in individual software routines for the case when the site has *not* set up a value for a given parameter. Entity, then, is implemented as a variable pointer.

A given parameter may occur for a variety of entities. In fact, you frequently need to obtain the value of a parameter by following an entity "chain." For example, the **Add Orders** menu a CPRS user sees may be defined at various levels. Initially, a site generally creates a custom **Add Orders** menu. Later, hospital locations may each build a custom menu that more specifically meets their needs. Individual users may also have their own **Add Orders** menus. If no site configuration has been done, the **Add Orders** menu exported with OE/RR is used. So, when OE/RR needs to display an **Add Orders** menu, a chain is followed that looks first to see if the user has their own menu. Next, the current location is checked, followed by the site. Finally, if no values exist, the software default menu is used.

In the PARAMETER DEFINITION (#8989.51) file, a multiple lists which entities are valid with a given parameter. These entities are also assigned a precedence, so that it is possible to write functions that will "chain" through entities until a value is found, using the proper sequence.

27.3 Description

Patch XT*7.3*26 contains a developer toolset that allows creation of software parameters in a central location. Integration Agreements (IAs) 2263 and 2336 define the supported entry points for this application. Kernel Patch XU*8.0*201 allows KIDS to transport the parameters.

Parameter Tools is a generic method of handling parameter definition, assignment, and retrieval. A parameter can be defined for various entities where an entity is the level at which you want to allow the parameter defined (e.g., software level, system level, division level, location level, user level, etc.). A developer can then determine in which order the values assigned to given entities are interpreted.

27.4 Definitions

The following are some basic definitions used by Parameter Tools:

- Entity
- Parameter
- Instance
- Value
- Parameter Template

27.4.1 Entity

An entity is a level at which you can define a parameter. The entities allowed are stored in the PARAMETER ENTITY (#8989.518) file. Kernel Toolkit patches maintain entries in this file. Table 55 lists the allowable parameter entries:

Table 55: Parameter Entities

Entity Prefix	Message	Points To File
PKG	Package	PACKAGE (#9.4)
SYS	System	DOMAIN (#4.2)
DIV	Division	INSTITUTION (#4)
SRV	Service	SERVICE/SECTION (#49)
LOC	Location	HOSPITAL LOCATION (#44)
TEA	Team	TEAM (#404.51)
CLS	Class	USR CLASS (#8930)
USR	User	NEW PERSON (#200)
BED	Room-Bed	ROOM-BED (#405.4)
OTL	Team (OE/RR)	OE/RR LIST (#100.21)

Entity Prefix	Message	Points To File
DEV	Device	DEVICE (#3.5)

Package (PKG), as an entity, allows the software defaults to be handled the same way as other parameters rather than hard-coded.

System (SYS), Division (DIV), Location (LOC), and User (USR) are frequent entries in existing software parameter files (or additions to the NEW PERSON [#200] file).

Service (SRV), Team (TEA), and Class (CLS) are referenced frequently by parameters that pertain to Notifications.

The process of exporting software using this kind of parameters file involves sending:

- Parameter definitions that belong to the software (entries in the PARAMETER DEFINITION [#8989.51] file).
- Actual parameter instances that point to the software (entries in the PARAMETERS [#8989.5] file that have an entity that matches the software).

All the other entries in the PARAMETERS (#8989.5) file (those that correspond to entities other than package [PKG]) would never be exported, as they are only valid for the system on which they reside.

27.4.2 Parameter

A parameter is the actual name under which values are stored. The name of the parameter *must* be:

- Namespaced
- Unique and start with two uppercase characters

Parameters can be defined to store the typical software parameter data (e.g., the default add order screen in OE/RR), but they can also be used to store graphical user interface (GUI) application screen settings a user has selected (e.g., font or window width). With each parameter, a more readable display name can also be defined. When a parameter is defined, the entities that may set that parameter are also defined. The definition of parameters is stored in the PARAMETER DEFINITION (#8989.51) file.

27.4.3 Instance

An instance is a unique value assigned to an entity/parameter combination. For most parameters, there will only be one instance, that is, instance does *not* apply and is simply set to **1**.

However, a parameter can be multi-valued—it can have more than one instance. More than one value can be assigned to the parameter as it relates to a specific entity. For example, lab collection times at a division. For a single entity (division in this case), multiple collection times may exist. Each collection time would be assigned a unique instance.

A parameter is *not* considered multi-valued if it can apply to several entities, but for each entity only one value of the parameter exists. For example, "maximum days for a lab order" can be set for every location in the hospital. However, since there is only one value for each location, "maximum days for a lab order" is *not* multi-valued.

When a parameter that is multi-valued is defined, the instance can be defined as any of the following:

- Numeric
- Date/Time
- Pointer
- Set Of Codes
- Free Text
- Yes/No

The validating logic for an instance is defined the same way as for a value.

27.4.4 Value

A value can be assigned to every parameter for the entities allowed in the parameter definition. Values are stored in the PARAMETERS (#8989.5) file. Fields in the PARAMETERS (#8989.5) file map to DIR fields. DIR is used to validate the data. Values can be any of the following:

- Numeric
- Date/Time
- Pointer
- Set Of Codes
- Free Text
- Yes/No
- Word-processing Type

27.4.5 Parameter Template

A Parameter template is similar to an Input template. It contains a list of parameters that can be entered through an input session (e.g., an option). Templates are stored in the PARAMETER TEMPLATE (#8989.52) file. Entries in this file *must* also be namespaced.

Table 56 lists the two Input templates for adding parameter definitions:

Table 56: Templates—Parameter Tools

Template	Description
XPAR SINGLE VALUED CREATE	For adding/editing parameters that will be single valued
XPAR MULTI VALUED CREATE	For adding/editing parameters that will be multiple valued

27.5 Why Use Parameter Tools?

The reason a developer would use Parameter Tools is to allow a hierarchical designation of a parameter value. Thus, rather than many parameters that exist now, which are just for the system level or just for a particular clinic, Parameter Tools allows you to define:

- Different levels at which the parameter can be set.
- In what priority the values are used.

Take, for example, setting up a default order menu for a person. Each facility may have a default order menu for their primary care clinicians. Each division may have one that is slightly different if their practices vary enough. For each location, they may set up a different order menu so that users working in a cardiology clinic get a different set of possible orders than those in a dermatology clinic. And there may be reasons to give one specific person a different order menu because they are authorized to prescribe additional medications, because they tend to practice in a different flow, or for other reasons. It's one parameter, but it allows the parameter to be set for multiple entities (at multiple levels). Those entities are defined in the IA, but can include package (PKG, which only developers should set—these are default export values), system (SYS, whole medical facility), division (DIV), location (LOC), room-bed (BED), team (TEA), provider, etc.

The PARAMETER DEFINITION (#8989.51) file defines what entities are allowed to be used for a parameter and in which order they are resolved (individual takes precedence over location takes precedence over division takes precedence over system which takes precedence over package). Sometimes you would want to create defaults for your medical center, but allow users in a certain area to customize what they see and do for their particular role.

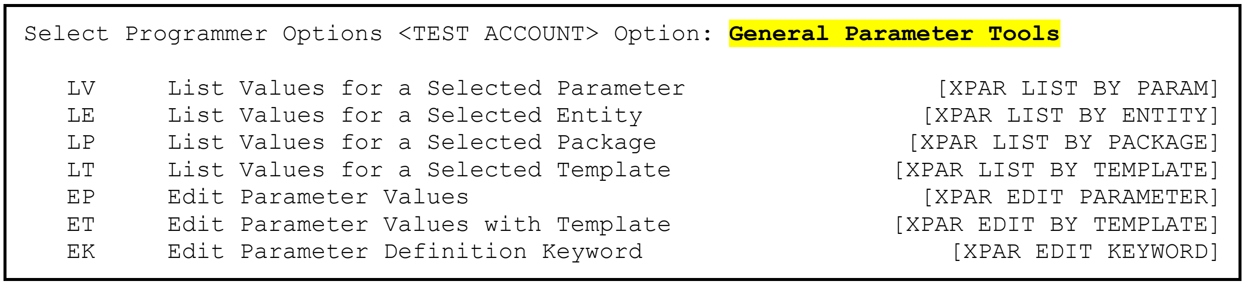
XPAR finds the appropriate value based on the parameter definitions and settings that may exist. This way, the developer does *not* need to look at multiple different location or person files to determine how the software should operate.

With integrations, this is even more important because it allows facilities to integrate; however, at the same time, continue some business practices based on parameters set at the division level rather than at the system level.

27.6 General Parameter Tools Menu

The **General Parameters Tools** [XPAR MENU TOOLS] menu (Figure 361), which is located on the **Programmer Options** [XUPROG menu; locked with the XUPROG security key], provides general purpose options for managing and editing parameters.

Figure 361: General Parameters Tools Menu [XPAR MENU TOOLS]



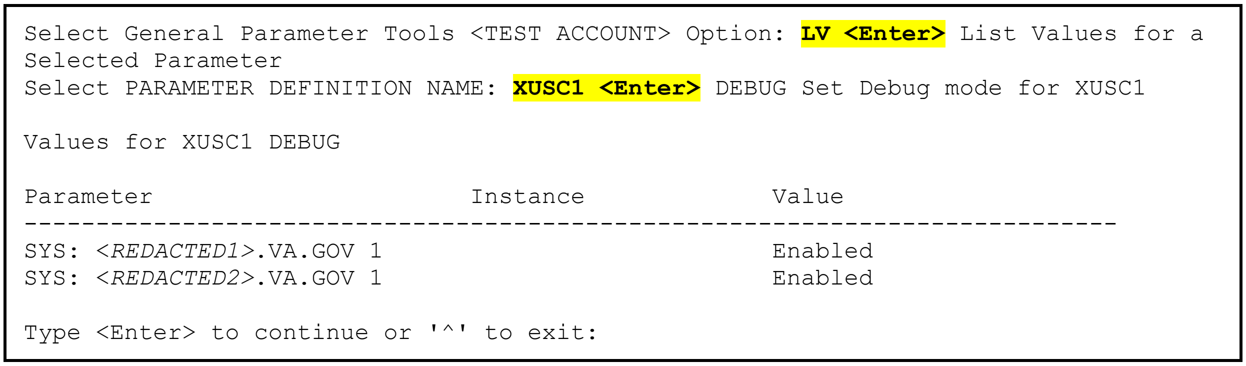
```
Select Programmer Options <TEST ACCOUNT> Option: General Parameter Tools

   LV     List Values for a Selected Parameter                [XPAR LIST BY PARAM]
   LE     List Values for a Selected Entity                  [XPAR LIST BY ENTITY]
   LP     List Values for a Selected Package                [XPAR LIST BY PACKAGE]
   LT     List Values for a Selected Template              [XPAR LIST BY TEMPLATE]
   EP     Edit Parameter Values                              [XPAR EDIT PARAMETER]
   ET     Edit Parameter Values with Template              [XPAR EDIT BY TEMPLATE]
   EK     Edit Parameter Definition Keyword                    [XPAR EDIT KEYWORD]
```

27.6.1 List Values for a Selected Parameter Option

The **List Values for a Selected Parameter** [XPAR LIST BY PARAM] option (Figure 362) prompts the user for a parameter defined in the PARAMETER DEFINITION (#8989.51) file and lists all value instances for that parameter.

The synonym for this option is "**LV**."

Figure 362: List Values for a Selected Parameter Option—Sample User Entries and Report



```
Select General Parameter Tools <TEST ACCOUNT> Option: LV <Enter> List Values for a
Selected Parameter
Select PARAMETER DEFINITION NAME: XUSC1 <Enter> DEBUG Set Debug mode for XUSC1

Values for XUSC1 DEBUG

Parameter                      Instance            Value
----------------------------------------------------------------------------
SYS: <REDACTED1>.VA.GOV 1                          Enabled
SYS: <REDACTED2>.VA.GOV 1                          Enabled

Type <Enter> to continue or '^' to exit:
```

27.6.2 List Values for a Selected Entity Option

The **List Values for a Selected Entity** [XPAR LIST BY ENTITY] option (Figure 363) prompts the user for the entry of an entity (e.g., location, user, etc.) and lists all value instances for that entity.

The synonym for this option is "**LE**."

Figure 363: List Values for a Selected Entity Option—Sample User Entries

```
Select General Parameter Tools <TEST ACCOUNT> Option: LE <Enter>  List Values for a
Selected Entity

Entities may be set for the following:

     10  User          USR    [choose from NEW PERSON]
     20  Team          TEA    [choose from ]
     30  Class         CLS    [choose from ]
     40  Location      LOC    [choose from HOSPITAL LOCATION]
     50  Service       SRV    [choose from SERVICE/SECTION]
     60  Division      DIV    [choose from INSTITUTION]
     70  System        SYS    [<REDACTED>.VA.GOV]
     80  Package       PKG    [choose from PACKAGE]
     90  Room-Bed      BED    [choose from ROOM-BED]
     100 Team (OE/RR)  OTL    [choose from OR TEST]
     110 Device        DEV    [choose from DEVICE]

Enter selection: 10 <Enter> User NEW PERSON
Select NEW PERSON NAME: XUUSER,ONE <Enter>  XUUSER,ONE       OEX         TECHNICAL
WRITER
```

Figure 364: List Values for a Selected Entity Option—Sample Report

```
Values for USR: XUUSER,ONE

Parameter                      Instance            Value
----------------------------------------------------------------------------
KMPD GUI OPTION GLOBAL LIST    1                   2
KMPD GUI OPTION ERROR LIST     1                   2
KMPD GUI OPTION ROUTINE SEARCH 1                   2
KMPD GUI OPTION LOOKUPS        1                   1
KMPD GUI OPTION CODE STATS     1                   2
KMPD GUI OPTION CODE EVALUATOR 1                   2
KMPD GUI OPTION TIMING MONITOR 1                   2
KMPD GUI OPTION ENVIRON CHECK  1                   2
KMPD GUI OPTION TOOLS PARAMS   1                   2
KMPD GUI OPTION ENVIRON SELECT 1                   SAGG
KMPD GUI OPTION RPT            1                   2~1

Type <Enter> to continue or '^' to exit:
```

27.6.3 List Values for a Selected Package Option

The **List Values for a Selected Package** [XPAR LIST BY PACKAGE] option (Figure 365) prompts the user for a package and lists all parameter values for the selected package.

The synonym for this option is "**LP**."

Figure 365: List Values for a Selected Package Option—Sample User Entries and Report

```
Select General Parameter Tools <TEST ACCOUNT> Option: LP <Enter> List Values for a
Selected Package
Select PACKAGE NAME: KERNEL <Enter> XU

Values for PKG: KERNEL

Parameter                      Instance             Value
----------------------------------------------------------------------------
XPAR TEST SET OF CODES         1                    Red
XUSNPI QUALIFIED IDENTIFIER    Individual_ID        VA(200,
XUSNPI QUALIFIED IDENTIFIER    Organization_ID      DIC(4,
XUSNPI QUALIFIED IDENTIFIER    Pharmacy_ID          PS(59,
XUSNPI QUALIFIED IDENTIFIER    Test_ID              TEST

Type <Enter> to continue or '^' to exit:
```

27.6.4 List Values for a Selected Template Option

The **List Values for a Selected Template** [XPAR LIST BY TEMPLATE] option (Figure 366) prompts the user for a parameter template. Depending on the definition of the template, additional information may be prompted for, and then the parameter values defined by the template are displayed.

The synonym for this option is "**LT**."

Figure 366: List Values for a Selected Template Option—Sample User Entries and Report

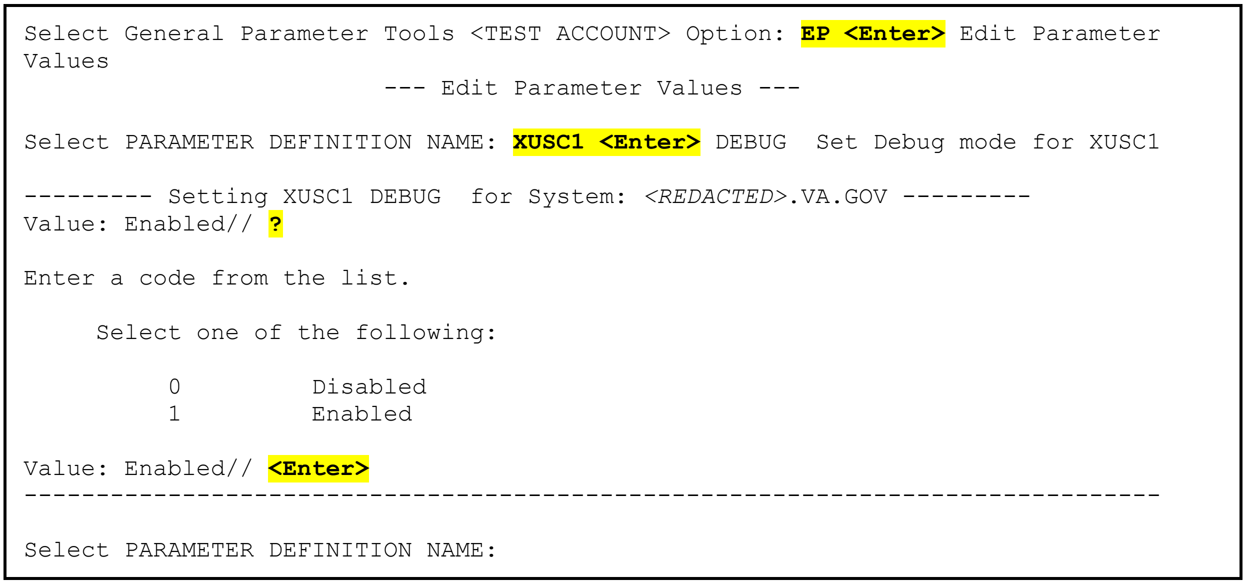
```
Select General Parameter Tools <TEST ACCOUNT> Option: LT <Enter> List Values for a
Selected Template
Select PARAMETER TEMPLATE NAME: OEX
     1   OEX TEST                       TEMPLATE FOR OEX TEST
     2   OEX TEST2                      TEMPLATE FOR OEX TEST2
     3   OEX TEST3                      TEMPLATE FOR OEX TEST3
CHOOSE 1-3: 1 <Enter>  OEX TEST  TEMPLATE FOR OEX TEST
Select INSTITUTION NAME: 13TH & MISSION <Enter> CA  D  662BU
Are you adding -1 as a new Instance? Yes// <Enter> YES

TEMPLATE FOR OEX TEST for Division: 13TH & MISSION, -1
------------------------------------------------------------------------------
THIS IS OEX TEST
------------------------------------------------------------------------------
Type <Enter> to continue or '^' to exit:
```

27.6.5 Edit Parameter Values Option

The **Edit Parameter Values** [XPAR EDIT PARAMETER] option (Figure 367) calls the low level parameter editor, which allows you to edit the values for every parameter. Normally, packages supply other means of editing parameters.

The synonym for this option is "**EP**."

Figure 367: Edit Parameter Values Option—Sample User Entries



```
Select General Parameter Tools <TEST ACCOUNT> Option: EP <Enter> Edit Parameter
Values
                         --- Edit Parameter Values ---

Select PARAMETER DEFINITION NAME: XUSC1 <Enter> DEBUG  Set Debug mode for XUSC1

--------- Setting XUSC1 DEBUG  for System: <REDACTED>.VA.GOV ---------
Value: Enabled// ?

Enter a code from the list.

     Select one of the following:

          0         Disabled
          1         Enabled

Value: Enabled// <Enter>
-------------------------------------------------------------------------------

Select PARAMETER DEFINITION NAME:
```

27.6.6 Edit Parameter Values with Template Option

The **Edit Parameter Values with Template** [XPAR EDIT BY TEMPLATE] option prompts the user for a parameter template, and then uses the selected template to edit parameter values.

The synonym for this option is "**ET**."

27.6.7 Edit Parameter Definition Keyword Option

The **Edit Parameter Definition Keyword** [XPAR EDIT KEYWORD] option allows a user to edit the keyword field in the PARAMETER DEFINITION (#8989.51) file.

The synonym for this option is "**EK**."

Figure 368: Edit Parameter Definition Keyword Option—Sample User Entries

```
Select General Parameter Tools <TEST ACCOUNT> Option: EK <Enter> Edit Parameter
Definition Keyword

Select PARAMETER DEFINITION NAME: XUSC1 <Enter> DEBUG  Set Debug mode for XUSC1
Select KEYWORD: DEVELOPER// ??
   DEVELOPER

        You may enter a new KEYWORD, if you wish
   This field provides a list of KEYWORDS that can be used for lookup of
   Parameter definitions.  It is suggested that each entry only have
   one word.

Select KEYWORD: DEVELOPER// DEBUG
  Are you adding 'DEBUG' as a new KEYWORD? No// Y <Enter> (Yes)
Select KEYWORD: <Enter>

Select PARAMETER DEFINITION NAME:
```

27.7 Example

The following procedure is a simple example of a way you might use the Parameter Tools. Suppose you needed a parameter that could be set as a default for the system (account) and also overridden for a given user. Previously, you had to add a field to a software site file (e.g., the KERNEL SYSTEM PARAMETERS [#8989.3] file) and then add a similar field to the NEW PERSON (#200) file. This situation is a perfect use of the Parameter Tools.

1. You need the equivalent to a data dictionary (DD) entry. Figure 369 goes into the PARAMETER DEFINITION (#8989.51) file. In this case, you need a Yes/No Set of Codes. So, this is what you set up:

Figure 369: Setting Up the PARAMETER DEFINITION (#8989.51) File

```
Name: XUS-XUP VPE
DISPLAY TEXT: Drop into VPE
MULTIPLE VALUED: N <Enter> No
VALUE DATA TYPE: Y <Enter> yes/no
VALUE HELP: Should XUP drop the user into the VPE environment?
Description...
PRECEDENCE: 1       ENTITY FILE: USER
PRECEDENCE: 2       ENTITY FILE: SYSTEM
```

NOTE: Figure 369 only shows the fields with the data necessary to set up the PARAMETER DEFINITION (#8989.51) file.

Figure 369 lists the order that values are looked for and returned. You want a USER value (File #200) if there is one; otherwise, a SYSTEM value (File #4.2). It also gives the entities that are allowed to have values of this data. In the place of SYSTEM, you could have used PACKAGE.

2. You can use **^XPAREDIT** to enter a value for your new parameter:

Figure 370: Use ^XPAREDIT to Enter Value for New Parameter

```
>D ^XPAREDIT

                 --- Edit Parameter Values ---

Select PARAMETER DEFINITION NAME: XUS-XUP VPE <Enter> Drop into VPE

XUS-XUP VPE may be set for the following:

     1   User          USR    [choose from NEW PERSON]
     2   System        SYS    [<REDACTED>.VA.GOV]

Enter selection: 2 <Enter> System   <REDACTED>.VA.GOV

----- Setting XUS-XUP VPE  for System: <REDACTED>.VA.GOV ---------
Value:  NO
...
```

3. How do you get this value out in your VistA application?

Figure 371: Get Value of New Parameter for VistA Application

```
>S X=$$GET^XPAR("USR^SYS","XUS-XUP VPE",1,"Q")  ;X will be null, 0 or 1.
```

- **First Parameter**—Value from **USR** (User / New Person) or **SYS** (System)
- **Second Parameter**—Name of the parameter: "**XUS-XUP VPE**"
- **Third Parameter**—Number of Instances. In this example, you only allow one instance (optional, defaults to **1** if *not* passed).
- **Fourth Parameter**—Format to return: Use "**Q**" to get the internal value.

4. Adding the parameter template with VA FileMan, Figure 372:

Figure 372: Adding a Sample Parameter Template

```
Select PARAMETER DEFINITION NAME: XUS-XUP VPE <Enter> Drop into VPE

NAME: XUS-XUP VPE// <Enter>
DISPLAY TEXT: Drop into VPE// <Enter>
MULTIPLE VALUED: No// <Enter>
INSTANCE TERM: <Enter>
VALUE TERM: <Enter>
PROHIBIT EDITING: <Enter>
VALUE DATA TYPE: yes/no// <Enter>
VALUE DOMAIN: <Enter>
VALUE HELP: Should XUP drop the user into the VPE environment.
VALUE VALIDATION CODE: <Enter>
VALUE SCREEN CODE: <Enter>
INSTANCE DATA TYPE: <Enter>
INSTANCE DOMAIN: <Enter>
INSTANCE HELP: <Enter>
INSTANCE VALIDATION CODE: <Enter>
INSTANCE SCREEN CODE: <Enter>
DESCRIPTION:
  1> This parameter controls if a user when exiting XUP is dropped into
  2> VPE or right to the ">" prompt.
EDIT Option: <Enter>
Select PRECEDENCE: 2// <Enter>
  PRECEDENCE: 2// <Enter>
  ENTITY FILE: SYSTEM// <Enter>
Select PRECEDENCE: <Enter>
```

Glossary

Term	Definition
Alpha Testing	In VA terminology, Alpha testing is when a VistA test software application is running in a site's account.
Auto Menu	An indication to Menu Manager that the current user's menu items should be displayed automatically. When AUTO MENU is *not* in effect, the user *must* enter a question mark at the menu's select prompt to see the list of menu items.
Beta Testing	In VA terminology, Beta testing is when a VistA test software application is running in a Production account.
Capacity Management	The process of assessing a system's capacity and evaluating its efficiency relative to workload in an attempt to optimize system performance. Kernel provides several utilities.
Checksum	A numeric value that is the result of a mathematical computation involving the characters of a routine or file.
Cipher	A system that arbitrarily represents each character as one or more other characters. (See also: ENCRYPTION.)
Common Menu	**SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). Options on this menu are available to all users. Entering two question marks (**??**) at the menu's select prompt displays any SECONDARY MENU OPTIONS (#203) available to the signed-on user along with the **Common** menu options available to all users.
Compiled Menu System (^XUTL Global)	Job-specific information that is kept on each CPU so that it is readily available during the user's session. It is stored in the **^XUTL** global, which is maintained by the menu system to hold commonly referenced information. The user's place within the menu trees is stored, for example, to enable navigation via menu jumping.
Computed Field	This field takes data from other fields and performs a predetermined mathematical function (e.g., adding two columns together). You do *not*, however, see the results of the mathematical function on the screen. Only when you are printing or displaying information on the screen do you see the results for this type of field.
DEA	Drug Enforcement Administration.
Device Handler	The Kernel module that provides a mechanism for accessing peripherals and using them in controlled ways (e.g., user access to printers or other output devices).

Term	Definition
DIFROM	VA FileMan utility that gathers all software components and changes them into routines (**namespaceI*** routines) so that they can be exported and installed in another VA FileMan environment.
Double Quote (“)	A symbol used in front of a **Common** option’s menu text or synonym to select it from the **SYSTEM COMMAND OPTIONS** [XUCOMMAND] menu (aka **Common** menu). For example, the five character string **“TBOX** selects the User’s Toolbox [XUSERTOOLS] **Common** option.
DR String	The set of characters used to define the **DR** variable when calling VA FileMan. Since a series of parameters may be included within quotes as a literal string, the variable’s definition is often called the **DR** string. To define the fields within an edit sequence, for example, the developer may specify the fields using a **DR** string rather than an INPUT template.
DUZ(0)	A local variable that holds the FILE MANAGER ACCESS CODE (#3) field of the signed-on user.
Encryption	Scrambling data or messages with a cipher or code so that they are unreadable without a secret key. In some cases encryption algorithms are one directional, that is, they only encode and the resulting data *cannot* be unscrambled (e.g., Access and Verify codes).
EPCS	Drug Enforcement Administration (DEA) Electronic-Prescribing of Controlled Substances (ePCS).
File Access Security System	Formerly known as Part 3 of the Kernel Inits. If the File Access Security conversion has been run, file-level security for VA FileMan files is controlled by Kernel’s File Access Security system, *not* by VA FileMan Access codes (i.e., FILE MANAGER ACCESS CODE [#3] field in the NEW PERSON [#200] file).
Forced Queuing	A device attribute indicating that the device can only accept queued tasks. If a job is sent for foreground processing, the device rejects it and prompt the user to queue the task instead.
Go-Home Jump	A menu jump that returns the user to the primary menu presented at signon. It is specified by entering two carets (**^^**) at the menu’s select prompt. It resembles the “Rubber-band Jump” but *without* an option specification/name following the carets.
Help Processor	A Kernel module that provides a system for creating and displaying online documentation. It is integrated within the menu system so that help frames associated with options can be displayed with a standard query at the menu’s select prompt.
Host File Server (HFS)	A procedure available on layered systems whereby a file on the host system can be identified to receive output. It is implemented by the Device Handler’s HFS device type.

Term	Definition
INIT	Initialization of an software application. **INIT*** routines are built by VA FileMan's DIFROM and, when run, recreate a set of files and other software components.
Jump	In VistA applications, the Jump command allows you to go from a particular field within an option to another field within that same option. You can also Jump from one menu option to another menu option without having to respond to all the prompts in between. To jump, type a caret (**^**) and then type the name of the field or option to which you wish to jump. (See also GO-HOME JUMP, PHANTOM JUMP, RUBBER-BAND JUMP, or UP-ARROW JUMP.)
Jump Start	A logon procedure whereby the user enters the "Access code;Verify code;option" to go immediately to the target option, indicated by its menu text or synonym. The jump syntax can be used to reach an option within the menu trees by entering "accesscode;verifycode;option".
Kermit	A standard file transfer protocol. It is supported by Kernel and can be set up as an alternate editor.
Manager Account	A UCI that holds vendor shared routines.
Menu Cycle	The process of first visiting a menu option by picking it from a menu's list of choices and then returning to the menu's select prompt. Menu Manager keeps track of information (e.g., the user's place in the menu trees) according to the completion of a cycle through the menu system.
Menu Manager	The Kernel module that controls the presentation of user activities (e.g., menu choices or options). Information about each user's menu choices is stored in the Compiled Menu System, the **^XUTL** global, for easy and efficient access.
Menu System	The overall Menu Manager logic as it functions within the Kernel framework.
Menu Template	An association of options as pathway specifications to reach one or more final destination options. The final options *must* be executable activities and *not* merely menus for the template to function. Any user can define user-specific MENU templates via the corresponding **Common** option.
Menu Trees	The menu system's hierarchical tree-like structures that can be traversed or navigated, like pathways, to give users easy access to various options.
PAC	**P**rogrammer **A**ccess **C**ode. An optional user attribute that can function as a second level password into programmer mode.
Part 3 of the Kernel Init	See FILE ACCESS SECURITY SYSTEM.

Term	Definition
Pattern Match	A preset formula used to test strings of data. Refer to your system's M Language Manuals for information on Pattern Match operations.
Phantom Jump	Menu jumping in the background. Used by the menu system to check menu pathway restrictions.
Primary Menus	The list of options presented at signon. Each user *must* have a PRIMARY MENU OPTION (#201) in order to sign on and reach Menu Manager. Users are given primary menus by system administrators. This menu should include most of the computing activities the user needs Value is stored in the PRIMARY MENU OPTION (#201) field in the NEW PERSON (#200) file.
Programmer Access	Privilege to become a developer on the system and work outside many of the security controls of Kernel. Accessing programmer mode from Kernel's menus requires having the at-sign security code (**@**), which sets the variable **DUZ(0)=@**.
Protocol	An entry in the PROTOCOL (#101) file. Used by the Order Entry/Results Reporting (OE/RR) software to support the ordering of medical tests and other activities. Kernel includes several protocol-type options for enhanced menu displays within the OE/RR software.
Queuing	Requesting that a job be processed in the background rather than in the foreground within the current session. Kernel's TaskMan module handles the queuing of tasks.
Queuing Required	An option attribute that specifies that the option *must* be processed by TaskMan (the option can only be queued). The option can be invoked and the job prepared for processing, but the output can only be generated during the specified time periods.
Resource	A method that enables sequential processing of tasks. The processing is accomplished with a RES device type designed by the application developer and implemented by system administrators. The process is controlled via the RESOURCE (#3.54) file.
Rubber-Band Jump	A menu jump used to go out to an option and then return, in a bouncing motion. The syntax of the jump is two carets (**^^**, uppercase-6 on most keyboards) followed by an option's menu text or synonym (e.g., ^^Print Option File). If the two carets are *not* followed by an option specification, the user is returned to the primary menu. (See also: GO-HOME JUMP.)
Scheduling Options	A way of ordering TaskMan to run an option at a designated time with a specified rescheduling frequency (e.g., once per week).

Term	Definition
Scroll/No Scroll	The **Scroll/No Scroll** button (also called Hold Screen) allows the user to “stop” (No Scroll) the terminal screen when large amounts of data are displayed too fast to read and “restart” (Scroll) when the user wishes to continue.
Secondary Menu Options	Options assigned to individual users to tailor their menu choices. If a user needs a few options in addition to those available on the primary menu, the options can be assigned as secondary options. To facilitate menu jumping, secondary menus should be specific activities, *not* elaborate and deep menu trees. Values are stored in the SECONDARY MENU OPTION (#203) field in the NEW PERSON (#200) file.
Secure Menu Delegation (SMD)	A controlled system whereby menus and security keys can be allocated by people other than system administrators (e.g., application coordinators) who have been so authorized. SMD is a part of Menu Manager.
Server Option	An entry in the OPTION (#19) file. An automated mail protocol that is activated by sending a message to the server with the “**S.server**” syntax. A server option’s activity is specified in the OPTION (#19) file and can be the running of a routine or the placement of data into a file.
Signon/Security	The Kernel module that regulates access to the menu system. It performs a number of checks to determine whether access can be permitted at a particular time. A log of signons is maintained.
Special Queueing	An option attribute indicating that TaskMan should automatically run the option whenever the system reboots.
Spooler	An entry in the DEVICE (#3.5) file. It uses the associated operating system’s spool facility, whether it’s a global, device, or host file. Kernel manages spooling so that the underlying OS mechanism is transparent. In any environment, the same method can be used to send output to the spooler. Kernel subsequently transfers the text to a global for subsequent despooling (printing).
Synonym	A field in the OPTION (#19) file. Options can be selected by their menu text or synonym. (See also: MENU TEXT.)
TaskMan	The Kernel module that schedules and processes background tasks (also called Task Manager).
Timed Read	The amount of time Kernel waits for a user response to an interactive **READ** command before starting to halt the process.
Up-Arrow Jump	In the menu system, entering a caret (**^**; sometimes referred to as an up-arrow) followed by an option specification/name accomplishes a jump to the target option without needing to take the usual steps through the menu pathway.

Term	**Definition**
XINDEX	A Kernel utility used to verify routines and other M code associated with a software application. Checking is done according to current ANSI MUMPS standards and VistA programming standards. This tool can be invoked through an option or from direct mode (>**D ^XINDEX**).
Z Editor (**^%Z**)	A Kernel tool used to edit routines or globals. It can be invoked with an option, or from direct mode after loading a routine with >**X ^%Z**.
ZOSF Global (**^%ZOSF**)	The Operating System File—a manager account global distributed with Kernel to provide an interface between VistA software and the underlying operating system. This global is built during Kernel installation when running the manager setup routine (**ZTMGRSET**). The nodes of the global are filled-in with operating system-specific code to enable interaction with the operating system. Nodes in the **^%ZOSF** global can be referenced by VistA application developers so that separate versions of the software need *not* be written for each operating system.

REF: For a list of commonly used terms and definitions, see the OIT Master Glossary VA Intranet Website.

For a list of commonly used acronyms, see the VA Acronym Lookup Intranet Website.

Index