



**ISS DEAVA PUBLIC KEY
INFRASTRUCTURE (PKI)
PILOT PROJECT**

SUPPLEMENT TO PATCH DESCRIPTION

Patches XU*8.0*283 and 288

June 2003

Department of Veterans Affairs
VistA Health Systems Design & Development (HSD&D)
Infrastructure and Security Services (ISS)

Revision History

Documentation Revisions

The following table displays the revision history for this document. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author
04/01/02	1.0	Initial ISS DEA/VA PKI Pilot Project (i.e., Kernel Patch XU*8.0*54) documentation creation.	Thom Blom and Wally Fort, Oakland, CA OIFO
10/30/02	2.0	Updated documentation based on developer input.	Thom Blom and Wally Fort, Oakland, CA OIFO
05/08/03	3.0	Changed patch references from XU*8.0*54 to XU*8.0*283 and added new error codes.	Thom Blom and Wally Fort, Oakland, CA OIFO
06/17/03	4.0	Modified document with updated diagrams, and minor formatting changes. Added new references to Kernel Patch XU*8.0*288, two new options, one new mail group, and made minor content updates based on other developer input.	Thom Blom and Wally Fort, Oakland, CA OIFO
10/22/03	4.1	Changed references from VistA Pharmacy's "Controlled Substances software" to "Outpatient Pharmacy software" throughout this document.	Thom Blom, Oakland, CA OIFO; Linda Hebert, Bay Pines, FL OIFO
01/27/05	4.2	<p>Reviewed document and edited for the "Data Scrubbing" and the "PDF 508 Compliance" projects.</p> <p>Data Scrubbing—Changed all patient/user TEST data to conform to HSD&D standards and conventions as indicated below:</p> <ul style="list-style-type: none"> • The first three digits (prefix) of any Social Security Numbers (SSN) start with "000" or "666." • Patient or user names are formatted as follows: KRNPATIENT,[N] or KRNUSER,[N] respectively, where the N is a number written out and incremented with each new entry (e.g., KRNPATIENT, ONE, KRNPATIENT, TWO, etc.). • Other personal demographic- 	Thom Blom, Oakland, CA OIFO

Revision History

		<p>related data (e.g., addresses, phones, IP addresses, etc.) were also changed to be generic.</p> <p>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).</p>	
--	--	--	--

Table i: Documentation revision history

Patch Revisions

Because this is a pilot project, the only associated patches are Kernel Patches XU*8.0*283 and 288. For a complete list of patches released with this software in the future, please refer to the Patch Module on FORUM.

Contents

Revision History	iii
Figures and Tables	ix
Acknowledgements.....	xi
Orientation	xiii
1. User Manual Information.....	1-1
Introduction	1-1
Purpose.....	1-2
Scope.....	1-3
Architecture Broad Overview Diagram.....	1-4
PKI Administrative Tasks	1-5
Introduction.....	1-5
Required Administrative Components and Processes.....	1-5
Administrative Tasks Overview Diagram.....	1-9
PKI Software Signing Functionality	1-10
Introduction.....	1-10
Step-By-Step Signing Procedures.....	1-10
Signing Functionality Overview Diagram	1-14
PKI Software Verification Functionality.....	1-15
Introduction.....	1-15
Step-By-Step Verification Procedures.....	1-15
Verification Functionality Overview Diagram	1-17
PKI Verification Server Process Diagram	1-18
2. Programmer Manual Information.....	2-1
Application Program Interfaces (APIs)	2-1
Controlled Subscription References	2-1
IXuDigSigS—Digital Signing COM API.....	2-1
\$\$STORESIG^XUSSPKI—PKI Data Storage API	2-5
\$\$VERIFY^XUSSPKI—Digital Signature Verification API.....	2-6
Supported References	2-7
\$\$DEA^XUSER()—Drug Enforcement Agency (DEA) Number API	2-7

3. Technical Manual Information3-1

- Implementation and Maintenance 3-1
 - Implementation 3-1
 - Maintenance 3-2
- Routines 3-4
- Global and File List 3-5
 - Global 3-5
 - Files 3-6
 - Fields 3-6
- Exported Options 3-7
 - Options—*With* Parents 3-7
 - Options—*Without* Parents 3-10
- Archiving and Purging 3-11
- Callable Routines 3-12
- External Interfaces 3-13
 - Hardware Interfaces 3-13
 - Software Interfaces 3-13
 - Communications Interfaces 3-13
- External Relations 3-14
 - Software Requirements 3-14
 - Dependencies 3-14
 - Integration Agreements (IA) 3-14
- Internal Relations 3-17
 - Namespace 3-17
 - File Numbers 3-18
- Software-wide Variables 3-19
- Software Product Security 3-20
 - Mail Groups 3-20
 - Remote System(s) 3-20
 - Archiving and Purging 3-20
 - Interfacing 3-21
 - Digital Signature(s) 3-21
 - Menu(s)/Option(s) 3-22
 - Security Key(s) 3-22

File Security 3-22

References..... 3-23

Official Policies 3-23

Glossary Glossary-1

Appendix A—API Error Management A-1

Index Index-1

Contents

Figures and Tables

Table i: Documentation revision history	iv
Table ii: Documentation symbol descriptions	xiii
Figure 1-1: Architecture Broad Overview diagram	1-4
Figure 1-2: Sample dialog box when asked to save console settings	1-8
Figure 1-3: Administrative Tasks Overview diagram.....	1-9
Figure 1-4: Signing Functionality Overview diagram	1-14
Figure 1-5: Verification Functionality Overview diagram	1-17
Figure 1-6: Verification Server Process diagram.....	1-18
Table 3-1: List of routines exported with the ISS DEA/VA PKI Pilot Project.....	3-4
Table 3-2: List of files used by the ISS DEA/VA PKI Pilot Project	3-6
Table 3-3: List of files used by the ISS DEA/VA PKI Pilot Project	3-6
Table 3-4: Menu options <i>with</i> a parent exported with the ISS DEA/VA PKI Pilot Project	3-7
Figure 3-1: Institution DEA# edit option example	3-8
Figure 3-2: Kernel PKI Parameter Edit option example.....	3-9
Table 3-5: Menu option <i>without</i> a parent exported with the ISS DEA/VA PKI Pilot Project.....	3-10
Table 3-6: Callable routines for the ISS DEA/VA PKI Pilot Project—Alphabetized by entry point.....	3-12
Table 3-7: File and global information for the ISS DEA/VA PKI Pilot Project.....	3-18
Table 3-8: Menu options exported with the ISS DEA/VA PKI Pilot Project.....	3-22
Table 3-9: File security for the ISS DEA/VA PKI Pilot Project.....	3-22
Table A-1: PKI API error codes and their resolutions.....	A-4

Figures and Tables

Acknowledgements

The Drug Enforcement Agency (DEA)/Department of Veterans Affairs (VA) Public Key Infrastructure (PKI) Pilot Project Team consists of the following Infrastructure and Security Services (ISS) personnel:

- ISS Program Director—Larry Weldon
- ISS PKI Project Manager—Dan Soraoka
- Lead Developer—Wally Fort
- Second Developer—Joel Ivey
- Consulting Developers—Mike Meighan
- Project Planner—Laura Rowland
- Software Quality Assurance (SQA)—Minao Murphy
- Technical Writer—Thom Blom

The ISS DEA/VA PKI Pilot Project Team would like to thank the following sites/organizations/personnel for their assistance in reviewing and/or testing the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) software and documentation (names within teams are listed alphabetically):

- Chief Management/Clinical Information—Jeff Ramirez
- Clinical Information—Ruth Anderson
- Director Health Data Systems (HDS)—Cynthia Kindred
- National Project Manager for PKI—Suzette Holston
- VistA Data Systems and Integration (VDSI)—Leigh Hurst, Kornel Krechoweckyj, John Kupecki, Catherine Pfeil, and Cameron Schlehber
- Enterprise Strategy Emerging Technologies—Dan Maloney
- Computerized Patient Record System (CPRS) Development Team—Tana Defa, Cynthia Kindred, Sheri Kreuz, Cary Malmrose, and Steve Monson
- Pharmacy Development Team—Mohamed Anwer, Luanne Barron, Stanley Brown, Teresa R. Evans, Valerie Howell, Shannon Templeton, and Eric Williamson
- IV&V
- Drug Enforcement Agency (DEA)—Sharon K. Partlo and Vickie Seeger
- Performance Engineering Corporation (PEC)—Steve Bruck, Tom Casey, Jason Mohler, Zarbana Noori, Gus Orogas, Mike Patnode, and Erik Pfeifer
- Maximus, Inc.—Paul Fleischman and Christy Mahler

Acknowledgements

Orientation

Manual Organization

This supplemental documentation to the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) is organized into three major sections based on the following functional divisions for inclusion into the Kernel V. 8.0 documentation at a later date:

1. User Manual Information
2. Programmer Manual Information
3. Technical Manual Information



The software and documentation provided with Kernel Patch XU*8.0*283 is only applicable through the DEA/VA PKI Project Development and Pilot phases. Both the software and documentation are *not* ready for national release at this time. Upon completion of the DEA/VA PKI Project Pilot phase, both the software and documentation will be re-evaluated and updated as necessary, as well as the performance of any necessary tasks (e.g., Integration Agreements).

How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of the ISS DEA/VA PKI Pilot Project and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VistA) and commercial off-the-shelf (COTS) software products.

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. The following table gives a description of each of these symbols:

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

Table ii: Documentation symbol descriptions

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

the Approved Application Abbreviations document and "N" represents the first name as a number spelled out and incremented with each new entry. For example, in Kernel (KRN) test patient and user names would be documented as follows: KRNPATIENT,ONE; KRNPATIENT,TWO; KRNPATIENT,THREE; etc.

- "Snapshots" of computer online displays (i.e., roll-and-scroll screen captures/dialogs) and computer source code are shown in a *non*-proportional font and enclosed within a box. Also included are Graphical User Interface (GUI) Microsoft Windows images (i.e., dialogs or forms).
 - User's responses to online prompts will be boldface type.
 - The "<Enter>" found within these snapshots indicate that the user should press the Enter or Return key on their keyboard.
 - Author's comments are displayed in italics or as "callout" boxes.



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

- Object Pascal code uses a combination of upper- and lowercase characters. All Object Pascal reserved words are in boldface type.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).

How to Obtain Technical Information Online

Exported file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.



Methods of obtaining specific technical information online will be indicated where applicable under the appropriate topic. Please refer to Chapter 3, "Technical Manual Information," in this manual for further information.

Help at Prompts

Kernel has online help and commonly used system default prompts. Users are strongly 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 Kernel.

To retrieve online documentation in the form of Help in Kernel:

- Enter a single question mark ("?") at a field/prompt to obtain a brief description. If a field is a pointer, entering one question mark ("?") displays the HELP PROMPT field contents and a list of choices, if the list is short. If the list is long, the user will be asked if the entire list should be displayed. A YES response will invoke the display. The display can be given a starting point by prefacing the starting point with an up-arrow ("^") as a response. For example, **^M** would start an alphabetic listing at the letter M instead of the letter A while **^127** would start any listing at the 127th entry.
- Enter two question marks ("??") at a field/prompt for a more detailed description. Also, if a field is a pointer, entering two question marks displays the HELP PROMPT field contents and the list of choices.
- Enter three question marks ("???) at a field/prompt to invoke any additional Help text that may be stored in Help Frames.

Obtaining Data Dictionary Listings

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



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

Assumptions About the Reader

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

- VistA computing environment (e.g., Kernel Installation and Distribution System [KIDS])
- VA FileMan data structures and terminology
- Microsoft Windows
- M programming language

It provides an overall explanation of the use, maintenance, and implementation of the ISS DEA/VA PKI Pilot Project Software and the changes contained in Kernel Patches XU*8.0*283 and 288. 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 home pages on the World Wide Web (WWW) for a general orientation to VistA. For example, go to the VHA OI Health Systems Design & Development (HSD&D) Home Page at the following Web address:

<http://vista.med.va.gov/>

Reference Materials

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

- *ISS DEA/VA PKI Pilot Project, Supplement to Patch Description (Patches XU*8.0*283 and 288)* (this manual)
- *Kernel Release Notes*
- *Kernel Installation Guide*
- *Kernel Systems Manual*
- *Kernel Programmer Manual*
- *Kernel Technical Manual*
- *Kernel Security Tools Manual*
- ISS Public Key Infrastructure (PKI) Home Page at the following Web address:

<http://vista.med.va.gov/pki/index.asp>

This site provides an overview/links for the ISS VA/DEA PKI pilot project.

- VA/DEA Public Key Infrastructure (PKI) Home Page at the following Web address:

<http://vaww.va.gov/techsvc/projects/vapkidea/vapkidea.asp>

This site provides an overview/links for the VA/DEA PKI pilot project.

- VA Public Key Infrastructure (PKI) Pilot Projects Home Page at the following Web address:

<https://vaww.webdev.med.va.gov/techsvc/projects/vapki.asp>

This site provides an overview/links for all VA PKI pilot projects.

- Kernel Home Page at the following Web address:

<http://vista.med.va.gov/kernel/index.asp>

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

VistaA 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 (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following Web address:

<http://www.adobe.com/>



For more information on the use of the Adobe Acrobat Reader, please refer to the *Adobe Acrobat Quick Guide* at the following Web address:

<http://vista.med.va.gov/iss/acrobat/index.asp>

VistaA documentation can be downloaded from the Health Systems Design and Development (HSD&D) VistaA Documentation Library (VDL) Web site:

<http://www.va.gov/vdl/>

VistA documentation and software can also be downloaded from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO <ftp.fo-albany.med.va.gov>
- Hines OIFO <ftp.fo-hines.med.va.gov>
- Salt Lake City OIFO <ftp.fo-slc.med.va.gov>
- Preferred Method <download.vista.med.va.gov>

This method transmits the files from the first available FTP server.



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

Orientation

1. User Manual Information

Introduction

The Pharmacy Benefits Management (PBM) Strategic Healthcare Group, in collaboration with the Drug Enforcement Administration (DEA), requested the development of the first Public Key Infrastructure (PKI) VistA pilot project, named Public Key Infrastructure for Electronic Prescriptions Pilot Project. The objective is to develop an electronic order entry system for Schedule II controlled substances using digital signatures. A Memorandum of Understanding (MOU) between the DEA and the Department of Veterans Affairs (VA) authorizes only specified sites to use the full functionality of this pilot, although additional functionality is included that will benefit non-pilot sites.

The following Kernel patches were created for this ISS project:

- XU*8.0*283—PKI Pilot Support
- XU*8.0*288—Updated DEA^XUSER API

In addition to the Kernel Patches XU*8.0*283 and 288, to make the DEA/VA PKI Pilot Project viable, modifications to the following VistA software applications were also identified:

- Computerized Patient Record System (CPRS) V. 22.0
- Controlled Substances (CS) V. 3.0
- National Drug File (NDF) V. 4.0
- Outpatient Pharmacy (OP) V. 7.0
- Pharmacy Data Management (PDM) V. 1.0



The modifications made to each of these VistA software applications in support of the DEA/VA PKI Pilot Project will *not* be discussed in this supplemental document. For more information, the reader must consult the appropriate documentation for each application listed.

This supplemental documentation is intended for use in conjunction with the DEA/VA PKI Pilot Infrastructure and Security Services (ISS) Project. It outlines the details of the work involved in the DEA and PKI Pilot ISS-related patches (i.e., XU*8.0*283 and 288) and gives guidelines on how the Application Program Interfaces (APIs) and Common Object Model (COM) objects can be used to digitally sign and verify Schedule II-V controlled substance prescription drug orders for a pilot project within the Veterans Health Administration (VHA).

This is the User Manual section of this supplemental documentation for the ISS DEA/VA PKI Pilot Project. It will be incorporated into the *Kernel V. 8.0 Systems Manual* at a later date.

The intended audience for this chapter is the application developers for the Computerized Patient Record System (CPRS) and Pharmacy software. However, it can also be helpful to others in Health Systems Design & Development (HSD&D), Information Resource Management (IRM), National VistA Support (NVS), and VistA Data Systems and Integration (VDSI).

Purpose

The purpose of this Infrastructure and Security Services (ISS) project is to design a Public Key Infrastructure (PKI) interface through Application Program Interfaces (APIs) in VistA's Kernel software for use by VistA's Computerized Patient Record System (CPRS) and Outpatient Pharmacy software. This is the first project that uses PKI services from within VistA. As a pilot project, its purpose for the DEA is to assist in the development and release of revised regulations to allow for the electronic transmission of digitally-signed prescriptions for Schedule II-V controlled substances. For VHA, it will help uncover issues regarding the use of PKI services in a VistA context.

Kernel Patch XU*8.0*283

Kernel Patch XU*8.0*283 contains the functionality required for the DEA/VA PKI Pilot Project and lays the foundations for the future implementation of the PKI once the DEA regulations are revised and published. It is installed with the PKI functionality disabled and will not have any impact until the PKI functionality is enabled. PKI functionality can be enabled via Computerized Patient Record System (CPRS) parameters at a site level and a user level.



Additional hardware and software is required from various sources before PKI can be implemented at a site. Do *not* enable the PKI parameters until all required software and hardware have been installed.

The DEA/VA PKI Pilot Project also requires a digitally signing COM API, a PKI data storage API, and a digital signature verification API. This patch provides these required APIs and also adds a new global, two new options, and a new mail group.



For information on APIs, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

For information on the new global, options, and mail group, please refer to Chapter 3, "Technical Manual Information," in this manual.

Kernel Patch XU*8.0*288

As required by the DEA/VA PKI pilot project, Kernel Patch XU*8.0*288 provides a new API to obtain the value stored in the DEA# field (#53.2) in the NEW PERSON file (#200). This patch also adds a new field, FACILITY DEA NUMBER (#52), to the INSTITUTION file (#4).

In addition, this patch updates the DEA API that was added with Kernel Patch XU*8.0*267. It allows a user IEN to be passed in instead of the current DUZ. Also, if the institution doesn't have a DEA# on file, a check is done to get the PARENT FACILITY and see if there is a DEA# for that entry.



For information on APIs, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

Scope

The scope of this project is to create one or more APIs that will give VistA application developers the ability to enable PKI functionality within VistA. The overall goal of the PKI Pilot is to develop an electronic order entry system for Schedule II-V controlled substances using digital signatures. Currently, DEA regulations require a Schedule II controlled substance prescription be in written form and bear the "wet" signature of the practitioner to prevent the diversion of pharmaceutical drugs.

The VA PKI Pilot Project overall architecture consists of three major parts:

- **PKI Administration**—For a complete description, please refer to the "PKI Administrative Tasks" topic that follows.
- **PKI Signing Functionality**—For a complete description, please refer to the "PKI Software Signing Functionality" topic that follows.
- **PKI Verification Functionality**—For a complete description, please refer to the "PKI Software Verification Functionality" topic that follows.



This is a PKI pilot project only and not for national release. At this time, it is not being written to specifically address any of the forthcoming Health Insurance Portability and Accountability Act (HIPAA) regulations with regards to PKI and security.

Architecture Broad Overview Diagram

The organizations responsible for and relationships between these three parts are illustrated below:

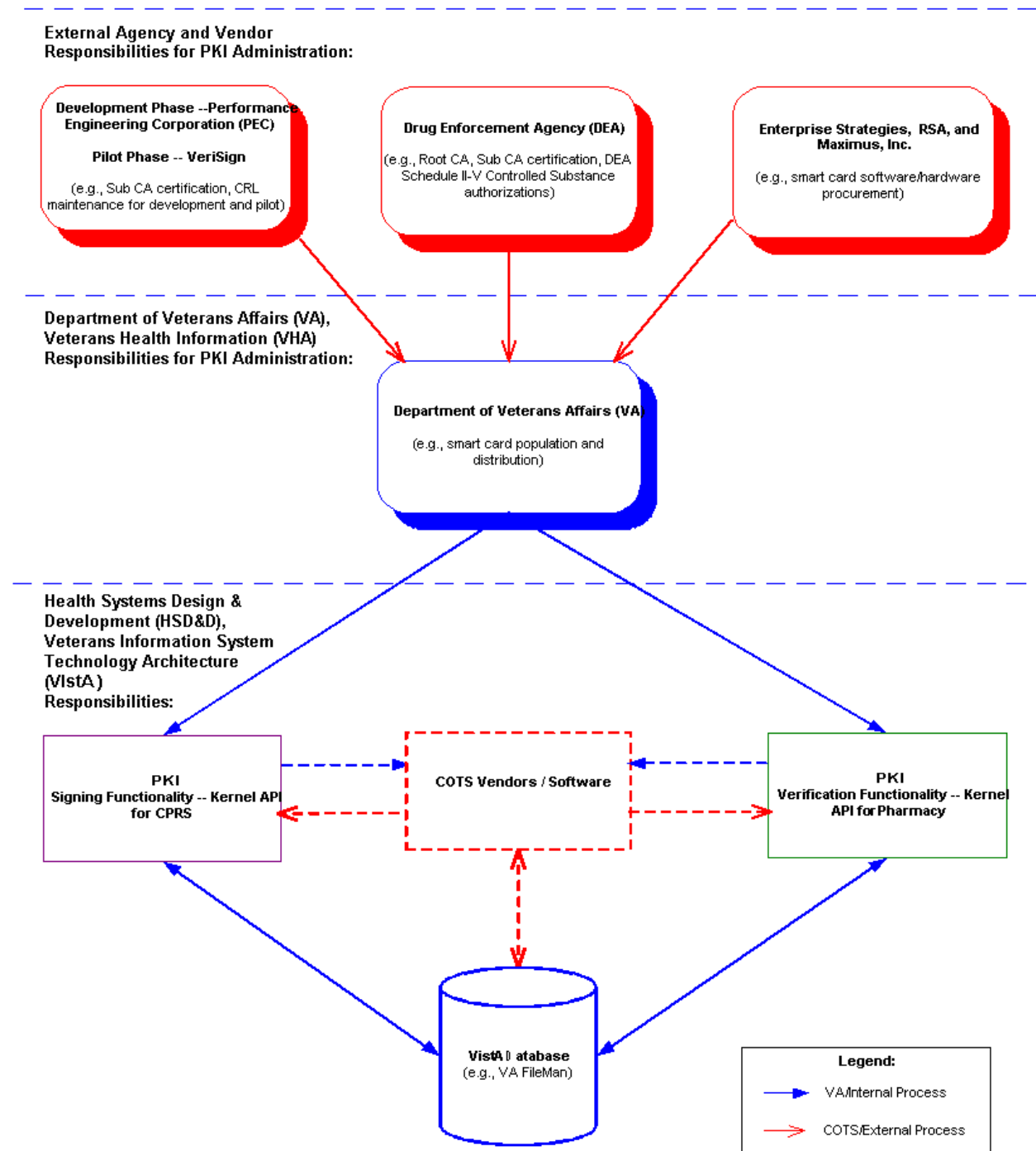


Figure 1-1: Architecture Broad Overview diagram

PKI Administrative Tasks

Introduction

The following administrative tasks are required for the Drug Enforcement Agency (DEA) and Public Key Infrastructure (PKI) Pilot Project and beyond. These tasks are required so that Department of Veterans Affairs (VA) practitioners will be granted the ability to digitally sign Schedule II-V Controlled Substances.

Required Administrative Components and Processes

The following describes the components and higher level processes that will be employed in order to administer PKI certificates.

Root Certification Authority

The Drug Enforcement Agency (DEA) is the Root Certification Authority (Root CA) for all Electronic Prescription for Controlled Substance (EPCS) Certificates (new and ongoing process). The Root CA is responsible for monitoring/maintaining the Sub CA's certification.

Subordinate Certification Authority

The Subordinate Certification Authority (Sub CA) for the VA must be established (new and ongoing process). The Sub CA hosts for the project phases are as follows:

- **Development Phase**—Performance Engineering Corporation (PEC)
- **Pilot Phase**—VeriSign

Certificate Revocation List (CRL)

A Certificate Revocation List (CRL) for the VA must be established (new and ongoing process). The Sub CA—Performance Engineering Corporation (PEC) n the CRL for the VA during the PKI development phase only (new arrangements will be required for the pilot and after). VeriSign will maintain the CRL for the pilot. The CRL will be create and maintained for the project phases as follows:

- **Development Phase**—Performance Engineering Corporation (PEC)
- **Pilot Phase**—VeriSign


DEA Authorization Process


The VA Practitioner must have a DEA number that authorizes them to prescribe Schedule II-V Controlled Substances (existing and ongoing process). The VA practitioner can obtain a DEA number in the following manner:

- Use a VA number (i.e., a local number that the site uses to track what users are using the site DEA number) with the VA facility DEA number.
- Apply to the Drug Enforcement Agency (DEA) for a DEA number that authorizes them to prescribe Schedule II-V Controlled Substances:

Certification Application Process

The VA Practitioner must submit an application for certification to the VA's Issuing Station and receive a certificate through the Subordinate Certificate Authority (new and ongoing process). If the certification is approved, the VA Practitioner gets a certificate. Otherwise, the VA Practitioner cannot prescribe Schedule II-V Controlled Substances.

- **VA Certificate Application**—The VA Practitioner applies for a PKI certificate through the VA's Issuing Station. They include their DEA number in the application. The VA's Issuing Station processes the VA Practitioner's application (new and ongoing process).
- **Sub CA Certificate Application**—The VA's Issuing Station does the following.
 - Contacts the Subordinate Certificate Authority (Sub CA) Web site passing verified certificate application data (new and ongoing process).
 - Receives Access code and password from the Sub CA for the VA Practitioner to download their certificate (new and ongoing process).
- **Smart Card Data Population**—The VA's Issuing Station populates a smart card (physical token) with the VA Practitioner's demographics (e.g., name, e-mail, organizational unit, etc.), and photo using a smart card Commercial Off-The-Shelf (COTS) interface that reads and writes data to a smart card (new and ongoing process).
 - G&D smart cards and SCM Microsystems smart card readers (Model SCR111) will be used for the PKI Pilot Project.
 -  Emerging Technologies group (under Enterprise Strategy) supplies and Maximus, Inc. formats the blank G&D smart cards. Emerging Technologies also provides smart card readers (exact number of cards and readers for the PKI Pilot to be determined).
 - RSA's Passage COTS software will be used for the PKI Pilot Project.

- **Smart Card Distribution**—The VA 's Issuing Station distributes the following to the VA Practitioner (new and ongoing process):
 - Newly populated smart card.
 - Personal Identification Number (PIN).
-  Access to the private key on the smart card requires the user enter their PIN every time it is accessed.
- Access code and password from Sub CA.
- **Smart Card Certificate Population**—The VA Practitioner uses the Access code and password, supplied by the VA's Issuing Station, to access the Sub CA's Web site to download the certificate. The Sub CA populates the smart card (physical token) with the VA Practitioner's certificate using a smart card Commercial Off-The-Shelf (COTS) interface that reads and writes data to a smart card (new and ongoing process):
 - G&D smart cards and SCM Microsystems smart card readers (Model SCR111) will be used for the PKI Pilot Project.
 - RSA's Passage COTS software will be used for the PKI Pilot Project.

The certificate includes various pieces of data stored on the smart card (e.g., version number, serial number, holder's name, country code, e-mail, issue date, valid through date, DEA number, private key, public key, and other certificate data).

- **Viewing Certificates on a Client Workstation**—To view any certificates stored on a client workstation do the following:
 - A. Go to the **Start** menu.
 - B. Select **Run**.
 - C. Type in **MMC** (upper- or lowercase).
 - D. Press the **OK** button—a **Console1** dialog will open.
 - E. Choose the **Console** menu option in the **Console1** dialog.
 - F. Select the **Add/Remove Snap-in...** menu item.
 - G. Press the **Add...** button in the **Add/Remove Snap-in** dialog—an **Add Standalone Snap-in** dialog will open.
 - H. Locate and highlight **Certificates** from the list presented in the **Add Standalone Snap-in** dialog.
 - I. Press the **Add** button—a **Certificates snap-in** dialog will open.
 - J. Leave the selection of "**My user account**" radio button marked and press the **Finish** button in the **Certificates snap-in** dialog.
 - K. Press the **Close** button in the **Add Standalone Snap-in** dialog.
 - L. Press the **OK** button in the **Add/Remove Snap-in** dialog.
 - M. You should now be able to review all of the certificates on the client workstation listed under the **Console Root** directory.

- N. When your review is complete, **Close** the **Console1** dialog. You will be asked to save your settings, as shown below:

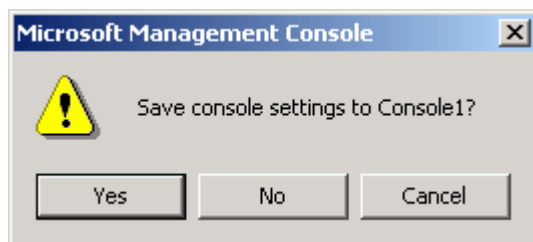


Figure 1-2: Sample dialog box when asked to save console settings

- O. Press the **Yes** button if you want to save the settings.
- P. Enter a unique name for the **.MSC** file (e.g., VIEWCERTS.MSC) and press the **Save** button. With saved settings, the next time you run MMC on this client workstation, you will just need to do the following: from the **Console** menu, select **Open...**, and then choose the **file name** you just saved from the list presented.

Certification Maintenance Process

The VA Practitioner is required to renew their DEA number with DEA every three years. The certificate will have to be replaced (i.e., the old certificate removed and a new certificate issued) at the same time (new and ongoing process).

Administrative Tasks Overview Diagram

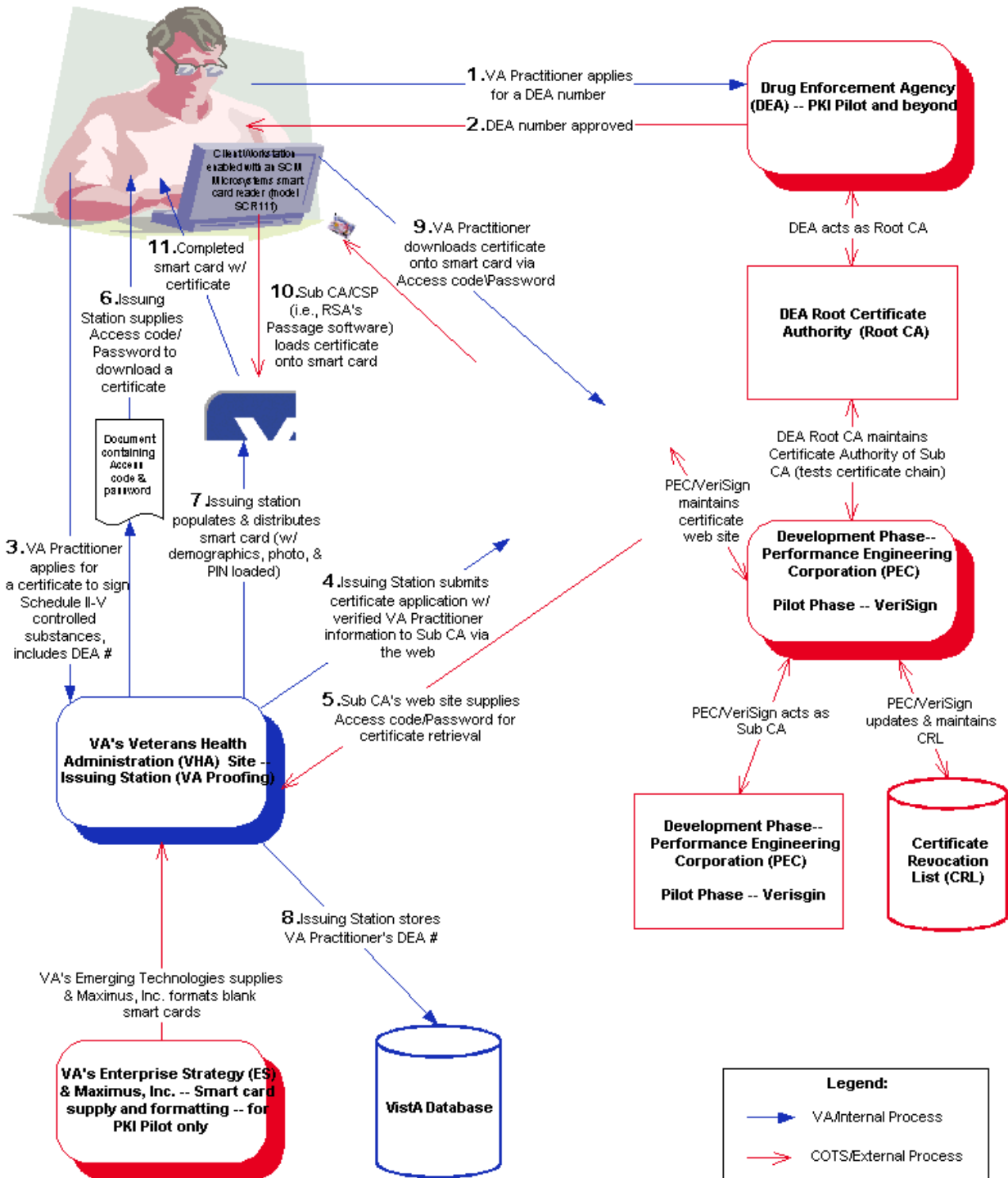


Figure 1-3: Administrative Tasks Overview diagram

PKI Software Signing Functionality

Introduction

Once a VA Practitioner examines a patient and determines that a Schedule II-V Controlled Substance should be prescribed, the prescription must be digitally signed.

Step-By-Step Signing Procedures

The following describes the higher-level step-by-step procedures/functionality that will be employed in order to digitally sign a prescription.

VA Practitioner Process

The VA Practitioner logs into VistA's Computerized Patient Records System (CPRS) software using a smart card-enabled workstation (new workstation requirement but existing and ongoing process):

- **Retrieve Patient Record**—VA Practitioner calls up the patient's record in CPRS (existing and ongoing process).
- **Prepare for Prescription Entry**—VA Practitioner chooses the "Orders Tab" (existing and ongoing process).
- **Enter Prescription Data**—VA Practitioner enters the prescription information for the Schedule II-V Controlled Substance being prescribed (existing and ongoing process).



For more information on VistA's CPRS software, please consult the documentation, located on the Web at:

<http://www.va.gov/vdl/Clinical.asp?appID=61>


VistA CPRS Software Process (Part I)

The VistA's CPRS software performs the following tasks:

- Retrieves the DEA number from VistA database (new and ongoing process).
- Calls the Infrastructure and Security Services (ISS) Development Team's Delphi-based Digital Signing Common Object Model (COM) using APIs to pass in the prescription block of data, DEA number, and DEA schedule (new and ongoing process).

Digital Signing COM Process

The Infrastructure and Security Services (ISS) Development Team's Delphi-based Digital Signing COM that is capable of digitally signing a block of data (e.g., a prescription drug order). In addition, this API will use COTS software that reads from and writes to smart cards (i.e., RSA's Passage software for the PKI Pilot). The API and COTS software perform the following tasks (new and ongoing process):

- **Check for Smart Card**—The ISS Digital Signing COM will check for a smart card using RSA's COTS software (new and ongoing process). The software prompts the VA Practitioner to place their smart card (physical token) into the smart card reader attached to the PC, if they haven't already done so.
 - **Locate DEA Certificate**—The ISS Digital Signing COM will search for certificate(s) on a smart card with the appropriate DEA extensions and a DEA number that matches the DEA number passed in by the CPRS software (new and ongoing process). This requires the use of Microsoft CryptoAPIs.
 - **Validate DEA Certificate**—The ISS Digital Signing COM performs validity checking (new and ongoing process):
 - Verifies certificate is valid for current date.
 - Verifies certificate has a DEA extension with matching DEA number.
 - **Digitally Sign a Block of Data**—The ISS Digital Signing COM will digitally sign the prescription drug order (new and ongoing process).
 - COTS software via the ISS Digital Signing COM prompts the VA Practitioner to enter their Personal Identification Number (PIN).
-  VA Practitioners must enter their Personal Identification Number (PIN) every time the private key on the smart card is accessed.
- The ISS Digital Signing COM hashes the prescription drug order block of data using a Federal Information Processing Standard (FIPS) algorithm (i.e., SHA hash, returning a 28-byte Base64 string).
 - The ISS Digital Signing COM digitally signs the hash with the DEA certificate information (see the "Smart Card Certificate Population" bullet under the "Certification Application Process" topic).

The Digital Signature is approximately 2,500 bytes. This is a full PKI digital signature with the private key and certificate on a smart card. The signature binary BLOB (Binary Large Object) is Base64 encoded for storage in Vista.

- **Return Digital Signature and Hash to Vista Calling Application**—The ISS Digital Signing COM has APIs to return the digital signature and hash to the calling application—Vista CPRS software (new and ongoing process).
- **Perform Error Processing**—The ISS Digital Signing COM will perform error processing as needed (new and ongoing process).

VistA CPRS Software Process (Part II)

The VistA's CPRS software performs the following tasks:

- Stores (long-term) the prescription block of data in VistA VA FileMan globals—Pharmacy and CPRS Order files/fields. The Pharmacy and CPRS Development Teams provide the storage capability (existing and ongoing process).
- Stores (long-term) the hash with the prescription block of data in a VistA VA FileMan global—Pharmacy file/field. The Pharmacy Development Team provides the storage capability (new and ongoing process).
- Calls the Information Infrastructure Service (ISS) Development Team's M-based PKI Data Storage API passing the digital signature, hash, and CRL URL associated with the certificate for long-term storage in VistA VA FileMan globals (files/fields). The ISS Development Team provides the storage capability (new and ongoing process).

VistA's CPRS software continues processing (existing and ongoing process).



For more information on VistA's CPRS software, please consult the documentation, located on the Web at:

<http://www.va.gov/vdl/Clinical.asp?appID=61>

PKI Data Storage API Process

The Infrastructure and Security Services (ISS) Development Team's M-based Data Storage API is a Kernel API that is capable of storing PKI-related data in the VistA database (i.e., digital signature and hash). The API performs the following tasks (new and ongoing process):

- **Store Digital Signature and Hash**—The ISS PKI Data Storage API stores (long-term) the digital signature and hash in the new PKI Digital Signature global, XUSHA (#8980.2). This global uses the hash as the primary key (.01). The ISS Development Team will provide the storage capability (new and ongoing process).
- **Perform Error Processing**—The ISS PKI Data Storage API will perform error processing as needed (new and ongoing process).

Prescription and Digital Signature Data

The following information needs to be included in the prescription and Digital Signature. This information was pulled from the DEA regulations (see Web site links below):

- Date of Prescription (Rx)
- Full name and address of the patient
- Drug name
- Drug strength
- Drug dosage form
- Drug quantity prescribed

- Direction for use
- Practitioner's name
- Practitioner's address
- Practitioner's (DEA) registration number
- Ink signature of practitioner (will be replaced by digital signature, a.k.a. PKI)



Links to the DEA regulations are provided below:

Current DEA regulations online:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/index.html>

DEA regulations dealing with prescriptions:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/2106cfrt.htm>

http://www.deadiversion.usdoj.gov/21cfr/cfr/1306/1306_05.htm

Section 1306.05 Manner of issuance of prescriptions.

Section 1306.05 Manner of issuance of prescriptions in the Title 21 Regulations and Codified CSA of the Code of Federal Regulations (CFR) specifically states the following:

"(a) All prescriptions for controlled substances shall be dated as of, and signed on, the day when issued and shall bear the full name and address of the patient, the drug name, strength, dosage form, quantity prescribed, directions for use and the name, address and registration number of the practitioner. A practitioner may sign a prescription in the same manner as he would sign a check or legal document (e.g., J.H. Smith or John H. Smith). Where an oral order is not permitted, prescriptions shall be written with ink or indelible pencil or typewriter and shall be manually signed by the practitioner. The prescriptions may be prepared by the secretary or agent for the signature of a practitioner, but the prescribing practitioner is responsible in case the prescription does not conform in all essential respects to the law and regulations. A corresponding liability rests upon the pharmacist who fills a prescription not prepared in the form prescribed by these regulations."¹

¹ DEA Web site (http://www.deadiversion.usdoj.gov/21cfr/cfr/1306/1306_05.htm): Section 1306.05 Manner of issuance of prescriptions.

Signing Functionality Overview Diagram

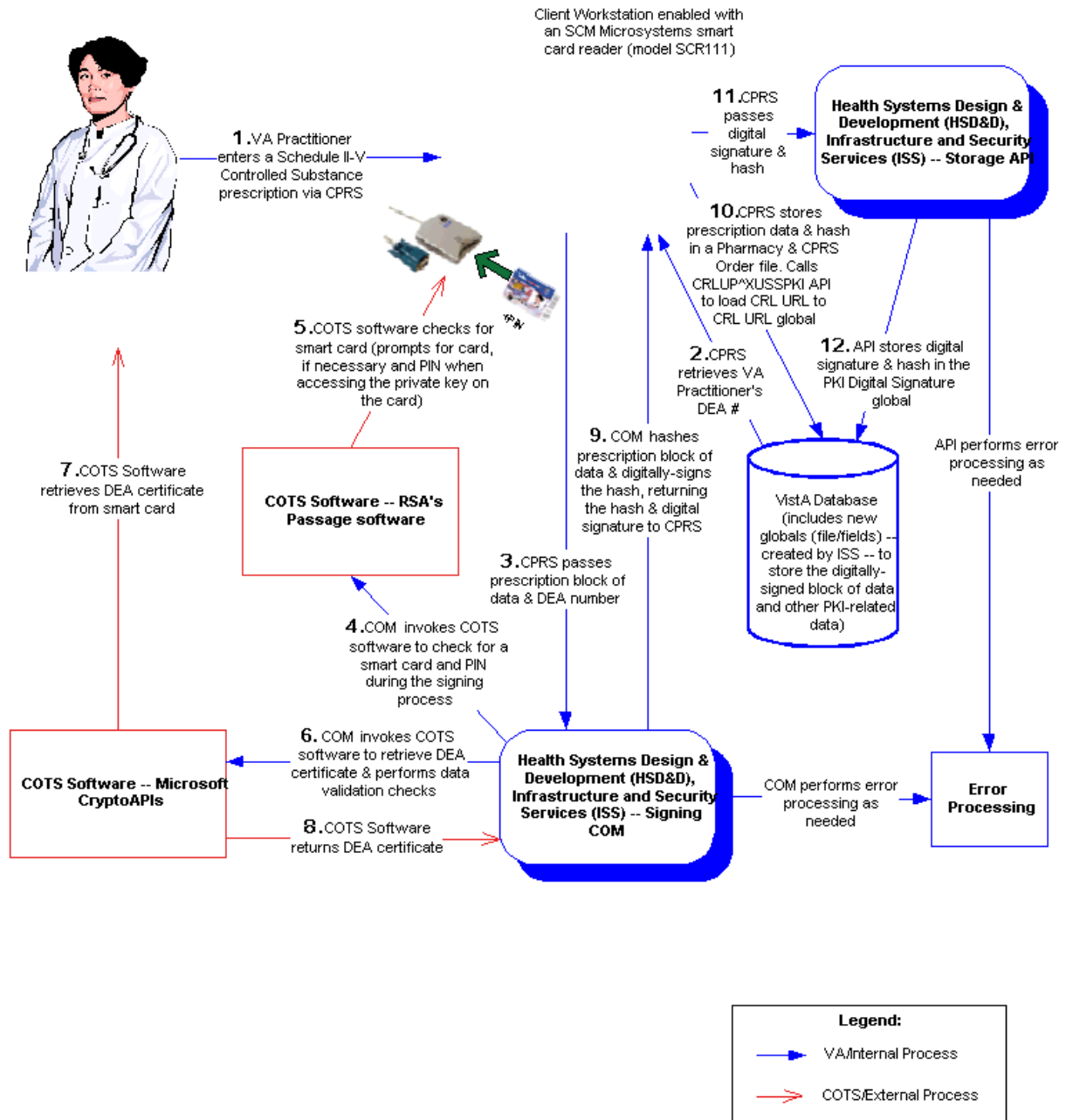


Figure 1-4: Signing Functionality Overview diagram

PKI Software Verification Functionality

Introduction

Once a VA Practitioner has prescribed a Schedule II-V Controlled Substance for a patient, both the prescription and the VA Practitioner's certification must be verified.

Step-By-Step Verification Procedures

The following describes the higher-level step-by-step procedures/functionality that will be employed in order to verify a digitally signed prescription:

VA Pharmacist Process

The VA pharmacist logs into VistA's Outpatient Pharmacy software in the roll-and-scroll environment (existing and ongoing process).



For more information on VistA's Outpatient Pharmacy software, please consult the documentation, located on the Web at:

<http://www.va.gov/vdl/Clinical.asp?appID=90>

VistA Pharmacy Software Process

The VistA's Outpatient Pharmacy software:

- Retrieves the hash and prescription block of data from the appropriate Pharmacy file in the VistA database (new/existing and ongoing process).
- Calls the Infrastructure and Security Services (ISS) Development Team's M-based Digital Signature Verification API passing the hash and prescription block of data (new and ongoing process).

Verifying Digital Signature API Process

The Infrastructure and Security Services (ISS) Development Team's M-based Digital Signature Verification API is a Kernel API that is capable of verifying a digitally signed block of data (e.g., a prescription drug order). In addition, this API will use COTS software. The API and COTS software perform the following tasks (new and ongoing process):

- **Retrieve Stored Certificate**—The ISS Digital Signature Verification API will retrieve the signature block of data (includes public key and certificate) from the new VistA long-term storage PKI Digital Signature global using the hash passed from the CPRS software as the primary key (new and ongoing process).
- **Connect to PKI Verification Server**—The ISS Digital Signature Verification API will open a connection to the PKI Verification Server (e.g., Windows NT/W2K server with Microsoft Crypto

APIs, running a Verification and CRL Retrieval service, new and ongoing process). The NT Service located on the PKI Verification Server automatically starts up on system boot and starts listening on a pre-defined port for connections.

- **Invoke COTS Software**—The ISS Digital Signature Verification API will invoke COTS software needed to verify the digitally signed block of data (i.e., digitally-signed prescription order). This requires the use of Microsoft CryptoAPIs (new and ongoing process).
- **Check the Trust Chain**—The ISS Digital Signature Verification API will check the trust chain (new and ongoing process):
 - **Sub CA**—Checks to see that the Sub CA is still certified by the Root CA.
 - **Certificate Revocation List (CRL)**—Checks to see that the certificate held by the VA Practitioner that prescribed the Schedule II-V Controlled Substance has not been revoked.
- **Return Response to VistA API**—The PKI Verification Server will pass the response to the ISS Digital Signature Verification API (new and ongoing process). The possible responses are:
 - **OK**—The prescription/digital signature has been verified. The VA Practitioner is certified at the time the prescription/digital signature is being verified and the Sub CA is still valid.
 - **Negative Number**—The prescription/digital signature has not been verified.
- **Disconnect from PKI Verification Server**—The ISS Digital Signature Verification API will close the connection to the PKI Verification Server (new and ongoing process).
- **Return Response to VistA Calling Application**—The ISS Digital Signature Verification API will pass the response from the PKI Verification Server back to the calling application—VistA Outpatient Pharmacy software (new and ongoing process). The possible responses are:
 - **OK**—The prescription has been verified. The VA Practitioner is certified at the time the prescription was written and the Sub CA is still valid.
 - **Negative Number**—The prescription has not been verified. VistA's Outpatient Pharmacy software will have to code for this situation.

VistA's Outpatient Pharmacy software continues processing the prescription order based on the response from the ISS Digital Signature Verification API (new/existing and ongoing process).

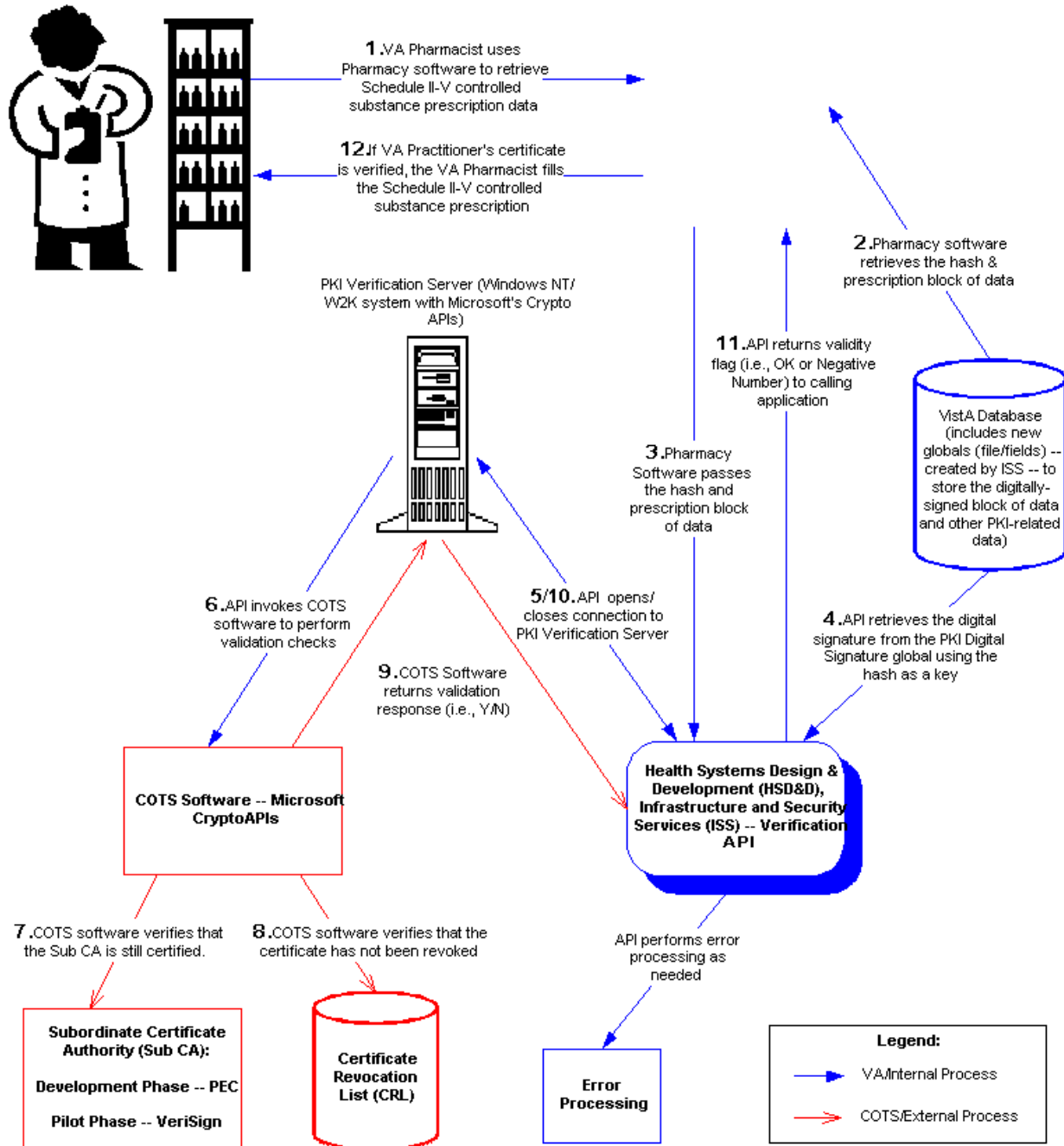


For more information on VistA's Outpatient Pharmacy software, please consult the documentation, located on the Web at:

<http://www.va.gov/vdl/Clinical.asp?appID=90>

- **Perform Error Processing**—The ISS Digital Signature Verification API will perform error processing as needed (new and ongoing process).

Verification Functionality Overview Diagram



NOTE: For more information on the PKI Verification Server, see the "PKI Verification Server Process Diagram"

Figure 1-5: Verification Functionality Overview diagram

PKI Verification Server Process Diagram

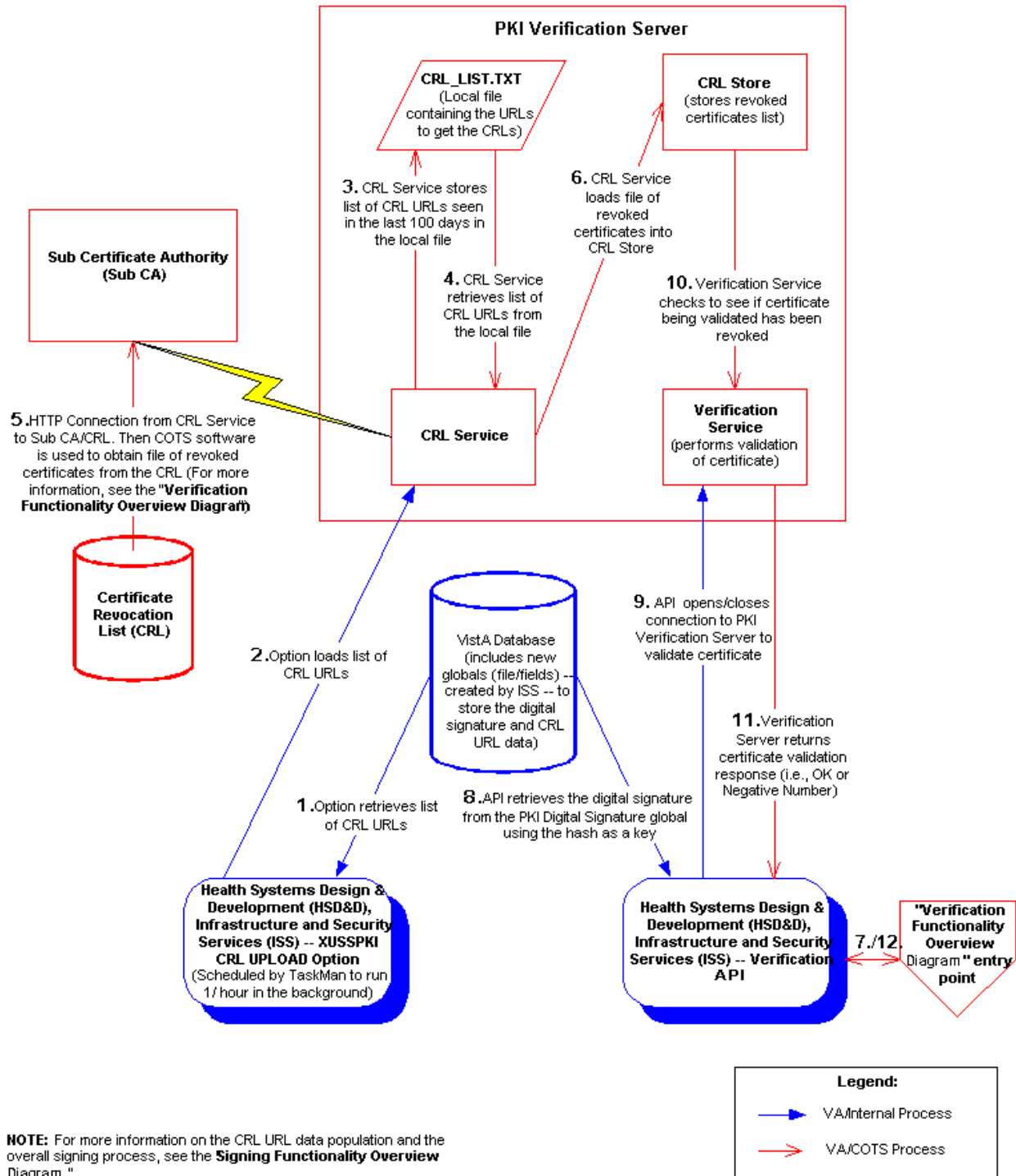


Figure 1-6: Verification Server Process diagram

2. Programmer Manual Information

This is the Programmer Manual section of this supplemental documentation for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). It will be incorporated into the *Kernel Systems Manual* at a later date.

The intended audience for this chapter is the application developers for the Computerized Patient Record System (CPRS) and Pharmacy software. However, it can also be helpful to others in Information Resource Management (IRM), National VistA Support (NVS), and VistA Data Systems and Integration (VDSI).

Application Program Interfaces (APIs)

This topic lists and describes the Kernel callable routines provided by the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). These calls are either Supported or Controlled Subscription IAs.



For a list of the Integration Agreements (IAs) related to the ISS DEA/VA PKI Pilot Project patch, please refer to the "Integration Agreements (IA)" topic in the "External Relations" topic in Chapter 3, "Technical Manual Information," in this manual.



Developer comments in the code are displayed in italics and blue font.

Controlled Subscription References

These are the Kernel Controlled Subscription Integration Agreements for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). They contain attributes/functions that *must* be controlled in their use. They will be recorded as a Controlled subscription Reference in the IA database on FORUM when this project is to be released nationally. Permission to use them is granted by the custodian package (i.e., Kernel software application) on a case-by-case basis.

IXuDigSigS—Digital Signing COM API

Category: Public Key Infrastructure (PKI)

Reference Type: Controlled Subscription

Integration Agreement Number: To be assigned.

Description:

This Common Object Module (COM) contains Kernel crypto APIs used for digitally signing a block of data. This COM object must be installed on each end-user client workstation that will be used for digitally signing data. These entry points are contained in the XuDigSigSC_TLB.PAS file.

Format:

```
//To start with you need to define a variable to hold the Object handle.
Crypto: IXuDigSigS;

//First create the crypto object.
crypto := CoXuDigSigS.Create;
//And see that we have a Smart Card and CSP.
//Throws an exception if it can't get the CSP or Smart Card Driver.
//After this call, the Reason property will have the CSP name.
Procedure GetCSP;

//Now put data into the object.
property DataBuffer: WideString writeonly
property UsrNumber: WideString writeonly
//This is set to the DEA schedule for the drug.
property DrugSch: WideString writeonly
//Set to true if we want to check for a DEA cert.
property DEAsig: WordBool
//Now to do the digital signature.
//The Passage software prompts the user if the card is not in the reader.
function SignData: WordBool;
//If it returns True then everything is OK and we collect the data.
property DEAINfo: WideString readonly
property HashValue: WideString readonly
property Signature: WideString readonly
property CrlUrl: WideString readonly
//
//If the SignData returned False then get the reason it failed.
property Reason: WideString readonly
```

Properties:

DataBuffer	The DataBuffer is a write-only wide-string property. It is used to store the string of pharmacy data that will be digitally signed.
UsrNumber	The UsrNumber is a write-only wide-string property. It is used to set the user's DEA registration number.
DrugSch	The DrugSch is a write-only wide-string property. Set this property to the DEA schedule number for the prescription drug being signed.
DEAsig	The DEAsig is a Boolean property. It is set to True if you want to check for a DEA registration number, which identifies a DEA certificate on the smart card. The default value is True. Set this property to False when you want to skip the check for the DEA registration number.

DEAInfo	The DEAInfo property is a read-only wide-string property. It is used to return the information from the DEA certificate found on the smart card.
HashValue	The HashValue property is a read-only wide-string property. It is used to return the hash used to digitally sign the block of data.
Signature	The Signature property is a read-only wide-string property. It is used to return the digital signature (i.e., digitally signed pharmacy data, DEA certificate information, and hash).
CrlUrl	The CrlUrl property is a read-only wide-string property. It is used to hold the CRL URLs to be used when verifying the digital signature.
Reason	The Reason property is a read-only wide-string property. If the SignData function fails for any reason, this property holds the text that describes the reason for the failure.

Methods:

GetCSP	This procedure is used to obtain the Cryptographic Service Provider (CSP) information and set the CSP name into the Reason property.
SignData	This function is called to digitally sign the block of data. If it returns True, then the signature process was successful. If it returns False, then the signature process failed and the function stores the reason for the failure in the Reason property.
Create	COM method to generate the link to the COM Server.

Example:

The following example demonstrates the use of the Digital Signing COM API:
 The following Delphi-based COM object would be called by CPRS to digitally sign data:

```

//Define this
var
crypto: IXuDigSigS;
//
Try
//First create the crypto object, this can be done earlier.
//The Passage software will prompt the user if the card is not in the reader.
    crypto := CoXuDigSigS.Create;
//Setup the CSP
    crypto.GetCSP;
//Then set the data into the object.
//Build a long string of the data that needs to be signed.
//This will have to be exactly re-created by Pharmacy when verifying the
//signature.
    crypto.Databuffer := databuffer; //This is the Pharmacy data to be signed
    crypto.UsrNumber := UsrDEANumber; //This is the users DEA Number from Vista
    crypto.UsrName := UserName; //This is the user name. Format unclear
    crypto.DrugSch := '2'; //This is set to a string that the users DEA schedule
//on the certificate must contain
    crypto.DEAsig := true; //Set to true if we want to check for a DEA cert.
    
```

```

//Now call to do work
//It will throw an error for many problems right now.
If crypto.SignData = false then
  begin
    FailureStatus := crypto.Reason;
    Crypto := nil; //Destroy the object
    exit;
  end;
//To get the DEA data from the certificate
DEAInfoStr := crypto.DEAInfo;
//To get the URL for the CRL dist point
CRLURL := crypto.CRLURL
CertDEAnum := $P(DEAInfoStr,"^",1);
//Need to collect the HASH
CertHash := crypto.HashValue;
//Need to collect the Signature
SignatureStr := crypto.Signature; //This will be from 1900 to 3600 bytes
/**Remember to free the COM object when done
crypto := nil;
//Now we need to send this down to Vista
//Send the SignatureStr and HashStr
//Send the Signature and Hash to new file
  with RPCBroker1 do begin
    RemoteProcedure := 'A6W1 PKI STORE SIG'; //RPC that can be used.
    Param[0].PType := literal;
    Param[0].Value := HashStr; //The HashStr to link the signature and data
    Param[1].PType := literal;
    Param[1].Value := IntToStr(len); //The length of the signature
    Param[2].PType := literal;
    Param[2].Value := '52'; //Data file (Pharmacy)
    with Param[3] do begin
      PType := list; //tells Broker to pass Mult
      while len >= IX do
        begin
          Mult[IntToStr(cnt)] := copy(wsig, ix, 240);
          inc(cnt);
          inc(ix, 240);
        end; //while
      end; //with param[3]
    end; //with RPCBroker
    RPCBroker1.Call; //execute RPC
  //Send the HashStr with the order to Pharmacy
  if pos('OK',RPCBroker1.Results.Text)=1 then
    begin
      canVerify := true;
      btnVerify.Enabled := true;
      //Make sure the data is saved.

      //Now save the URL's for the CRL's
      with RPCBroker1 do begin
        RemoteProcedure := 'A6W1 PKI STORE URL';
        Param[0].PType := literal;
        Param[0].Value := crypto.CrlUrl;
      end; //with
      RPCBroker1.Call;
    end; //end if

```



For a list of possible error codes associated with this COM/API, please refer to the "Appendix A—API Error Management" chapter in this manual.

\$\$\$STORESIG^XUSSPKI—PKI Data Storage API

Category: Public Key Infrastructure (PKI)

Reference Type: Controlled Subscription

Integration Agreement Number: To be assigned.

Description:

This M-based API stores the Digital Signature in the PKI DIGITAL SIGNATURE file (#8980.2).

Format:

```
$$$STORESIG^XUSSPKI(xu1,xu2,xu3,xu4,xu5)
```

Input Parameters:

xu1:	(required) This is the hash.
xu2:	(required) This is the data string length.
xu3:	(required) This is an array for the signature.
xu4:	(required) This is the DUZ of the signer.
xu5:	(required) This is the file that holds the data.

Output:

returns:	A "1" will be returned if the data was filed successfully. A "-1^message" is returned if an error occurred.
-----------------	--

Example:

The following example is an example of the PKI Data Storage M-based API:

```
SIG(RET,X1,X2,X3,X4) ;Store the signature. X1 is the hash
;X2 is the length of the array, X3 is an array for the sig
;X4 is the datafile
N Y1,Y2
S Y2=DUZ
S Y1=$$$STORESIG^XUSSPKI(X1,X2,.X4,Y2,X3)
S RET=$S(Y1=1:"OK",1:Y1)
```

\$\$VERIFY^XUSSPKI—Digital Signature Verification API

Category: Public Key Infrastructure (PKI)

Reference Type: Controlled Subscription

Integration Agreement Number: To be assigned.

Description:

This M-based API is called to verify the Digital Signature.

Format:

```
$$VERIFY^XUSSPKI(hash,$NA(in),[datesigned])
```

Input Parameters:

hash:	(required) The stored hash.
in:	(required) A Closed Root that points to the Pharmacy data.
datesigned:	(optional) The date the block of data was signed.

Output:

returns:	If the return value is "OK," then the digital signature has been successfully verified. Otherwise, there was an error and the digital signature has not been verified. The Pharmacy software will have to process the error.
-----------------	--

Example:

The following example demonstrates the use of Digital Signature Verification API:

```
N IN,OUT,HASH
S IN(1)=<PHARMACY DATA> ;This has to be the same as CPRS used
S IN(2)=<More pharmacy DATA> ;but can be in one or more nodes.
S HASH=<the stored hash>
;If the datesigned is not sent the current date will be used.
S S=$$VERIFY^XUSSPKI(HASH,$NA(IN),[DATESIGNED])
IF S<0 THEN W "SOME ERROR: ", $P(S,"^",2) Q
IF S="OK" THEN <D-SIG VERIFIED> Q
Q
```

Supported References

This is the Kernel Supported Integration Agreement for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). This API is open for use by any VistA application. It will be recorded as a Supported Reference in the IA database on FORUM when this project is to be released nationally.

\$\$DEA^XUSER()—Drug Enforcement Agency (DEA) Number API

Category: Public Key Infrastructure (PKI)

Reference Type: Supported

Integration Agreement Number: 2343

Description:

This extrinsic function returns a user's DEA number, if it exists in the DEA# field (#53.2) of the NEW PERSON file (#200). If the DEA# field value is null, the value returned depends on the optional FLAG input parameter (see "Input Parameters" below). This API allows a user IEN to be passed in to use in place of the current DUZ. Also, if the institution doesn't have a DEA# on file, a check is done to get the PARENT FACILITY and see if there is a DEA# for that entry.



This extrinsic function was originally released with Kernel Patch XU*8.0*267 and was updated via Kernel Patch XU*8.0*288.

Format:

```
$$DEA^XUSER([flag][,userien])
```

Input Parameters:

<p>flag:</p>	<p>(optional) This flag controls what is returned when the user does not have a value in the DEA# field (#53.2) of the NEW PERSON file (#200).</p> <p>Null or 0 This routine will check to see if the user has values in the VA# field (#53.3) of the NEW PERSON file (#200) and the (new) FACILITY DEA NUMBER field (#52) of the INSTITUTION file (#4). If values are found in both of those fields, this routine will return the following:</p> <p style="padding-left: 40px;">FACILITY DEA NUMBER field (#52)_"-"_VA# field(#53.3)</p> <p>1 This routine will check to see if the user has a value in the VA# field (#53.3) of the NEW PERSON file (#200). If a value is found in that field, this routine will return that field value. Otherwise, this routine returns an empty string.</p>
<p>userien:</p>	<p>(optional) This value can be used to get the DEA# of some user besides the one that signed in. In CPRS to check that a students teacher has the required DEA#.</p>

Output:

<p>returns:</p>	<p>Returns the DEA# field (#53.2) value or the value returned based on the (optional) Flag input parameter, see "Input Parameter" above.</p>
------------------------	--

Example 1:

This is the first example:

- IEN = "1000118"
- DEA# (#53.2) field = "AB1234567"
- FACILITY DEA NUMBER field (#52) = "VA7654321"
- VA# field (#53.3) = "789"

If the FLAG input parameter is null or "0", this API would return "AB1234567," since the user has a DEA#:

```
>S X=$$DEA^XUSER(0,1000118)
>W X
AB1234567
```

If the FLAG input parameter is "1", this API would return "AB1234567," since the user has a DEA#:

```
>S X=$$DEA^XUSER(1,1000118)
```

```
>W X
AB1234567
```

Example 2:

This is the second example:

- DEA# (#53.2) field = null
- FACILITY DEA NUMBER field (#52) = "VA7654321"
- VA# field (#53.3) = "789"

If the FLAG input parameter is null or "0", this API would return "VA7654321-789":

```
>S X=$$DEA^XUSER(0,)
```

```
>W X
VA7654321-789
```

If the FLAG input parameter is "1", this API would return "789":

```
>S X=$$DEA^XUSER(1,)
```

```
>W X
789
```

Example 3:

This is the third example:

- DEA# (#53.2) field = null
- FACILITY DEA NUMBER field (#52) = "VA7654321"
- VA# field (#53.3) = null

If the FLAG input parameter is null or "0", this API would return "":

```
>S X=$$DEA^XUSER(0,)
```

```
>W X
```

If the FLAG input parameter is "1", this API would return "":

```
>S X=$$DEA^XUSER(1,)
```

```
>W X
```

In both cases it returns an empty string.

3. Technical Manual Information

This is the Technical Manual section of this supplemental documentation for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). It will be incorporated into the *Kernel Technical Manual* at a later date.

This chapter will only address Infrastructure and Security Services (ISS) responsibilities to and relationship with the DEA/VA PKI Pilot Project, specifically Kernel Patches XU*8.0*283 and 288.

The intended audience for this chapter is Information Resource Management (IRM). However, it can also be helpful to others in Health Systems Design & Development (HSD&D), the Program Office, National VistA Support (NVS), and VistA Data Systems and Integration (VDSI).

Implementation and Maintenance

Implementation

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) is a Kernel Installation and Distribution System (KIDS) software release.

Kernel Patch Installation Instructions

Kernel Patches XU*8.0*283 and 288 can be installed at anytime. The installation time should not take more than a few minutes.



For specific installation instructions for the ISS DEA/VA PKI Pilot Project-related software (i.e., Kernel Patches XU*8.0*283 and 288), please refer to the Patch module on FORUM.



For specific software requirements and the minimum VistA software applications and patches that are required with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288), please refer to the "Software Requirements" topic under the "External Relations" topic that follows in this chapter.

Post Installation Instructions

The following are the post installation instructions for IRM personnel at pilot test sites:

- Download the XuDigSigSC.exe file from any of the NVS anonymous File Transfer Protocol (FTP) directories. This self-installing executable contains the COM object that is used by CPRS when digitally signing a prescription. It must be installed on all workstations that will be used to digitally sign a prescription.



For more information on the digitally signing COM object (API), please refer to the "IXuDigSigS—Digital Signing COM API" topic in the "Application Program Interfaces (APIs)" section in Chapter 2, "Programmer Manual Information" in this manual.

- Use TaskMan's Schedule/Unschedule Options option to schedule PKI CRL Upload option [XUSSPKI CRL UPLOAD] to run once an hour.



For more information on the Kernel PKI Parameter Edit option, please refer to the "Options—*Without Parents*" topic in the "Exported Options" section in this chapter.

- Use Kernel's PKI Parameter Edit option [XUSSPKI EDIT] to enter or edit the Internet Protocol (IP) address of the Verification Server.



For more information on the Kernel PKI Parameter Edit option, please refer to the "Options—*With Parents*" topic in the "Exported Options" section in this chapter.

- Use the Mail Group Edit option [XMEDITMG] to add users to the XUSSPKI CRL SERVER mail group.



For more information on the XUSSPKI CRL SERVER mail group, please refer to the "Mail Groups" topic in this chapter.



For any additional or more information on the post-installation instructions for the ISS DEA/VA PKI Pilot Project-related software (i.e., Kernel Patches XU*8.0*283 and 288), please refer to the Patch module on FORUM.

Memory Constraints

There are no special memory constraints, other than sites having sufficient space to allow for normal global growth.



For more information on the ^XUSSPKI global, please refer to the "Global" topic in the "Global and File List" section in this chapter.

Special Operations

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) does *not* require any special operations other than the normal backup and recovery operations.

Maintenance

Bulletins

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) does *not* generate any Bulletins.

Mail Groups

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) creates the following mail group:

XUSSPKI CRL SERVER

The XUSSPKI CRL SERVER mail group receives messages regarding problems with communication to the CRL Server.

IRM should use the Mail Group Edit option [XMEDITMG] to add the appropriate users to this mail group.

Routines

This section provides information related to all executable XU* routines exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). Do *not* delete any XU* routines.

All ISS DEA/VA PKI Pilot Project routines are prefixed with the namespace XU. The ISS DEA/VA PKI Pilot Project is composed of and exports seven executable routines.

The second line of these routines looks like:

```
<tab>;;8.0;KERNEL;<XU*8*283>;Jul 10, 1995
```



Other routine information, such as the Routine Size Histogram, the Routine %Index, etc., can be generated through the use of Kernel Utilities.

Routine	Description
XUSC1	M client/server routine. This routine is used in the connection process with the PKI Verification Server.
XUSC1C	M client/server routine. This routine is used in the connection process with the PKI Verification Server.
XUSC1S	M client/server routine. This routine is used in the connection process with the PKI Verification Server.
XUSC1S1	M client/server routine. This routine is used in the connection process with the PKI Verification Server.
XUSSPKI	M PKI routines. This routine obtains the signature and prescription data and stores it in a global.

Table 3-1: List of routines exported with the ISS DEA/VA PKI Pilot Project

Global and File List

This section contains information on all ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288)-related files and globals. This information includes: file numbers, file names, global location, data with file indicator, and file descriptions. ISS DEA/VA PKI Pilot Project file numbers range from 8980.2 to 8980.22.



File security access is described in the "File Security" topic in the "Software Product Security" section that follows in this chapter.

Other pertinent file information, such as data dictionaries and relations with other files, can be generated online through the use of VA FileMan utilities.

Global

Kernel Patch XU*8.0*283 creates the ^XUSSPKI global. Following national deployment, a site could see approximately 3,000 characters in this file for every Schedule II, III, and IV outpatient drug that is prescribed.



The data in the ^XUSSPKI global must be held for a minimum of two years.



The ^XUSSPKI global should be placed *prior* to installing Kernel Patch XU*8.0*283.

Files


VistA File Name & File Number	Global Location	Data W/ File?	Description
PKI DIGITAL SIGNATURES (#8980.2)	^XUSSPKI(8980.2,	No	This file stores the digitally signed prescription block of data.
PKI CRL URLS (#8980.22)	^XUSSPKI(8980.22,	No	<p>This file stores the list of Certificate Revocation List (CRL) Uniform Resource Locators (URLs). The PKI Verification Server retrieves this list and uses it in the verification process.</p> <p> An API will be required to connect to the CRL Service on the PKI Verification Server to send the list of CRL URLs. After the list has been received, the CRL Service contacts each URL via an HTTP connection to download the list of revoked certificates. On the PKI Verification Server, this list of revoked certificates is then imported into the CRL Store and is used by the Verification Service to determine whether or not the certificate in question is valid (i.e., not revoked).</p>

Table 3-2: List of files used by the ISS DEA/VA PKI Pilot Project

Fields

VistA Field Name & Field Number	File Location	Description
PKI SERVER ADDRESS field (#53.1)	KERNEL SYSTEM PARAMETERS file (#8989.3)	Kernel Patch XU*8.0*283—
FACILITY DEA NUMBER field (#52)	INSTITUTION file (#4)	Kernel Patch XU*8.0*288—

Table 3-3: List of files used by the ISS DEA/VA PKI Pilot Project

Exported Options

Options—*With Parents*

The following options are exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and XU*8.0&288) software. The options are assigned to the Kernel Management Menu [XUKERNEL]. The which is located on the Operations Management menu [XUSITEMGR], which is under the Systems Manager Menu [EVE].

Option Name	Option Title	Description
XU-INSTITUTION-DEA	Institution DEA# edit	Kernel Patch XU*8.0*288 added this option. This option is used to edit the Facility DEA number in the INSTITUTION file (#4).
XUSSPKI EDIT	Kernel PKI Parameter Edit	Kernel Patch XU*8.0*283 added this option. This option is used to enter or edit the Internet Protocol (IP) address of the Verification Server.

Table 3-4: Menu options *with* a parent exported with the ISS DEA/VA PKI Pilot Project

Institution DEA# edit	[XU-INSTITUTION-DEA]
------------------------------	-----------------------------

The Institution DEA# edit option [XU-INSTITUTION-DEA]] is used to edit the Facility Drug Enforcement Agency (DEA) number in the INSTITUTION file (#4).

The associated prompts and user responses when using the Institution DEA# edit option are shown below:

```
Select Operations Management Option: kernel Management Menu

      Edit Site IP lockout
      Enter/Edit Kernel Site Parameters
      Institution DEA# edit
      Institution Edit
      Kernel New Features Help
      Kernel Parameter File Edit
      Kernel PKI Parameter Edit

Select Kernel Management Menu Option: instit
  1  Institution DEA# edit
  2  Institution Edit
CHOOSE 1-2: 1 <Enter> Institution DEA# edit

Select INSTITUTION NAME: 662bu <Enter> 13TH & MISSION CA D 662BU
FACILITY DEA NUMBER: ?
      Answer with a DEA ID, must be 9 characters in length
FACILITY DEA NUMBER:
```

Here you must enter a valid DEA number.

Figure 3-1: Institution DEA# edit option example

Kernel PKI Parameter Edit	[XUSSPKI EDIT]
----------------------------------	-----------------------

The Kernel PKI Parameter Edit option [XUSSPKI EDIT] is used to enter or edit the Internet Protocol (IP) address of the Verification Server. This option updates the PKI SERVER ADDRESS field (#53.1) in the KERNEL SYSTEM PARAMETERS file (#8989.3)

The associated prompts and user responses when using the Kernel PKI Parameter Edit option are shown below:

```

Select Operations Management Option: kernel Management Menu

    Edit Site IP lockout
    Enter/Edit Kernel Site Parameters
    Institution DEA# edit
    Institution Edit
    Kernel New Features Help
    Kernel Parameter File Edit
    Kernel PKI Parameter Edit

Select Kernel Management Menu Option: kernel pki Parameter Edit
-----
                                Kernel PKI edit
    DOMAIN: NXT.KERNEL.FO-OAKLAND.MED.VA.GOV

    PKI Server: 10.?.??.???
    
```

After choosing the XUSSPKI EDIT, you are automatically placed into a ScreenMan form.

Here you would enter the IP address of the Verification Server configured at your site.

COMMAND: Press <PF1>H for help Insert

Figure 3-2: Kernel PKI Parameter Edit option example

Options—*Without Parents*

The following option is exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patch XU*8.0*283) software. This option is *not* assigned to any menu. The site System Manager should enter this option in the OPTION SCHEDULING file (#19.2) and schedule it to run once every hour.

Option Name	Option Title	Description
XUSSPKI CRL UPLOAD	PKI CRL Upload	This option calls CRLUP^XUSSPKI, which sends a list (string) of Uniform Resource Locators (URLs) for the Certificate Revocation Lists (CRLs) up to the PKI Verification Server. After the server accepts this list, it attempts to download the file pointed to by the URL for the certificate verification process. This option needs to be scheduled in TaskMan to run once per hour.

Table 3-5: Menu option *without* a parent exported with the ISS DEA/VA PKI Pilot Project



For more information on the PKI Verification Server process, please refer to the "PKI Verification Server Process Diagram" topic in Chapter 1, "User Manual Information," in this manual.

Archiving and Purging

There are no application-specific archiving procedures or recommendations for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288).

The PKI DIGITAL SIGNATURES file (#8980.2) has the potential to grow in size over time. However, for the ISS DEA/VA PKI Pilot Project this should not be an issue.



DEA regulations should first be consulted when determining whether or not any PKI-related data should be purged from the system.



Links to the DEA regulations are provided below:

Current DEA regulations online:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/index.html>

DEA regulations dealing with prescriptions:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/2106cfrt.htm>

http://www.deadiversion.usdoj.gov/21cfr/cfr/1306/1306_05.htm

Callable Routines

This topic lists the APIs exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288).

These callable entry points are described in Chapter 2, "Programmer Manual Information," in this manual. Please refer to the indicated topic under "For More Information" for more details about the calls in this documentation.

Entry Point	Brief Description	For More Information
IXuDigSigS Delphi-Kernel-Crypto COM API	This function digitally signs a prescription block of data.	See the "IXuDigSigS—Digital Signing COM API" topic in Chapter 2 in this manual.
\$\$STORESIG^XUSSPKI	This function stores the digitally signed prescription block of data.	See the "\$\$STORESIG^XUSSPKI—PKI Data Storage API" topic in Chapter 2 in this manual.
\$\$VERIFY^XUSSPKI	This function verifies a digitally signed prescription block of data.	See the "\$\$VERIFY^XUSSPKI—Digital Signature Verification API" topic in Chapter 2 in this manual.

Table 3-6: Callable routines for the ISS DEA/VA PKI Pilot Project—Alphabetized by entry point

External Interfaces

Hardware Interfaces

The ISS DEA/VA PKI Pilot Project APIs and files will reside on the standard hardware platforms employed by Department of Veterans Affairs healthcare facilities. These systems consist of DSM on OpenVMS or Caché on NT.



For a list of ISS DEA/VA PKI Pilot Project APIs, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

For a list of the ISS DEA/VA PKI Pilot Project files, please refer to the "Global and File List" topic in this chapter.

In addition, a separate PKI Verification Server is required. This server should be a Microsoft Windows NT/W2K server with Microsoft Crypto APIs, running a Verification and CRL Retrieval service.

The following COTS hardware products will be required with the ISS DEA/VA PKI Pilot Project:

- G&D Smart Cards
- SCM Microsystems Smart Card Readers (Model SCR111)

Software Interfaces

The following COTS hardware products will be required with the ISS DEA/VA PKI Pilot Project:

- RSA's Passage COTS Software



For more information on RSA, please refer to the RSA Home Page at the following Web address:

<http://www.rsa.com/>

Communications Interfaces

The XUSC1* routines communicate using a standard TCP/IP socket.



For a list of the XUSC1* routines, please refer to the "Routines" topic in this chapter.

External Relations

Software Requirements

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) requires a standard Vista operating environment in order to function correctly. Check your Vista environment for software applications and versions installed.

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) requires that *fully* patched versions of the following software:

- Kernel V. 8.0
- Kernel Toolkit V. 7.3
- VA FileMan V. 22.0
- MailMan V. 8.0

Dependencies

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) provides a common set of APIs for digitally signing and verifying Schedule II-V controlled substance prescriptions.



For more information on the APIs associated with the ISS DEA/VA PKI Pilot Project patches, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

Integration Agreements (IA)

Controlled Subscription References

The APIs associated with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) will be Supported and Controlled Subscription Integration Agreements (IAs). The Controlled Subscription IAs contain attributes/functions that *must* be controlled in their use. When this project is to be released nationally, they will be recorded as a Controlled subscription Reference in the IA database on FORUM. Permission to use them is granted by the custodian package (i.e., Kernel software application) on a case-by-case basis.



The software and documentation provided with Kernel Patch XU*8.0*283 is only applicable through the DEA/VA PKI Project Development and Pilot phases. Both the software and documentation are *not* ready for national release at this time. Upon completion of the DEA/VA PKI Project Pilot phase, both the software and documentation will be re-evaluated and updated as necessary, as well as the performance of any necessary tasks (e.g., Integration Agreements).



For more information on the APIs associated with the ISS DEA/VA PKI Pilot Project patches, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

General Instructions for Obtaining Integration Agreements

To obtain the current list of active IAs of which Kernel, which includes the ISS DEA/VA PKI Pilot Project, is a custodian:

- 1.) Sign on to the **FORUM** system.
- 2.) Select the **DBA menu [DBA]**.
- 3.) Select the **Integration Agreements Menu [DBA IA ISC]**.
- 4.) Select the **Custodial Package Menu [DBA IA CUSTODIAL MENU]**.
- 5.) Choose the **ACTIVE by Custodial Package option [DBA IA CUSTODIAL]**.
- 6.) Enter **KERNEL** at the "Select PACKAGE NAME:" prompt. You may have to further refine your choice, if presented with a list of similar named software applications.
- 7.) Choose the device to display the list of IAs.
- 8.) All current active IAs for which Kernel is custodian are listed.

To obtain detailed information on a specific integration agreement:

- 1.) Sign on to the **FORUM** system.
- 2.) Select the **DBA menu [DBA]**.
- 3.) Select the **Integration Agreements Menu [DBA IA ISC]**.
- 4.) Choose the **Inquire option [DBA IA INQUIRY]**.
- 5.) Enter the **integration agreement number of the IA you would like to display** (e.g., DBIA2171) at the "Select INTEGRATION REFERENCES:" prompt.
- 6.) Choose the device to display the list of IA.
- 7.) The full text of the requested IA will be displayed.

To obtain the current list of IAs to which Kernel, which includes the ISS DEA/VA PKI Pilot Project, is a subscriber:

- 1.) Sign on to the **FORUM** system.
- 2.) Select the **DBA menu [DBA]**.
- 3.) Select the **Integration Agreements Menu [DBA IA ISC]**.
- 4.) Select the **Subscriber Package Menu [DBA IA SUBSCRIBER MENU]**.
- 5.) Choose the **Print ACTIVE by Subscribing Package option [DBA IA SUBSCRIBER]**.
- 6.) Enter **KERNEL** (in uppercase) at the "START WITH SUBSCRIBING PACKAGE: FIRST/" prompt.
- 7.) Enter **KERNEL** (in uppercase) at the "GO TO SUBSCRIBING PACKAGE: LAST/" prompt.
- 8.) Choose the device to display the list of IAs.
- 9.) All current active IAs to which Kernel is a subscriber are listed.

Internal Relations

The PKI CRL Upload option [XUSSPKI CRL UPLOAD] is exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288):



For more information on the PKI CRL Upload option [XUSSPKI CRL UPLOAD] option, please refer to the "Options—*Without Parents*" topic in the "Exported Options" section in this chapter.

Namespace

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) uses Kernel's **XU** namespace. All routines and globals used in the ISS DEA/VA PKI Pilot Project begin with **XUSSPKI** or **XUSC1**.

File Numbers

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) file numbers and global locations are listed as follows:

File #	Global
8980.2	^XUSSPKI(8980.2,
8980.22	^XUSSPKI(8980.22,

Table 3-7: File and global information for the ISS DEA/VA PKI Pilot Project

The full Data Dictionaries for these files are being exported with ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288). The new file and field definitions will be transported in the KIDS transport global and installed at the site.



For more information on the ^XUSSPKI global, please refer to the "Global" topic in the "Global and File List" section in this chapter..

Software-wide Variables

There are *no* software-wide variables contained within the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288).

Software Product Security

Mail Groups

The following mail group is exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288):

XUSSPKI CRL SERVER



For more information on the XUSSPKI CRL SERVER mail group, please refer to the "Mail Groups" topic in the "Implementation and Maintenance" section in this chapter.

Remote System(s)

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) software contacts the PKI Verification Server and uses COTS software in order to check the CRL and trust chain stored on remote systems.

Archiving and Purging

There are *no* application-specific archiving procedures or recommendations for the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288).

The PKI DIGITAL SIGNATURES file (#8980.2) has the potential to grow in size over time. However, for the ISS DEA/VA PKI Pilot Project this should not be an issue.



DEA regulations should first be consulted when determining whether or not any PKI-related data should be purged from the system.



Links to the DEA regulations are provided below:

Current DEA regulations online:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/index.html>

DEA regulations dealing with prescriptions:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/2106cfrt.htm>

http://www.deadiversion.usdoj.gov/21cfr/cfr/1306/1306_05.htm

Interfacing

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) requires the use of the following specialized, non-VA products (hardware and software):

Hardware

The ISS DEA/VA PKI Pilot Project APIs and files will reside on the standard hardware platforms employed by Department of Veterans Affairs healthcare facilities. These systems consist of DSM on OpenVMS or Caché on NT.



For a list of ISS DEA/VA PKI Pilot Project APIs, please refer to the "Application Program Interfaces (APIs)" topic in Chapter 2, "Programmer Manual Information," in this manual.

For a list of the ISS DEA/VA PKI Pilot Project files, please refer to the "Global and File List" topic in this chapter.

In addition, a separate PKI Verification Server is required. This server should be a Microsoft Windows NT/W2K server with Microsoft Crypto APIs, running a Verification and CRL Retrieval service.

The following COTS hardware products will be required with the DEA/VA PKI Pilot Project:

- G&D Smart Cards
- SCM Microsystems Smart Card Readers (Model SCR111)

Software

The following COTS software products will be required with the DEA/VA PKI Pilot Project:

- RSA's Passage COTS Software



For more information on RSA, please refer to the RSA Home Page at the following Web address:

<http://www.rsa.com/>

Digital Signature(s)

The purpose of the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) is to create digital signatures for Schedule II-V controlled substance prescription orders.





For more information on the PKI and Digital Signature process, please refer to Chapter 1, "User Manual Information," in this manual.

Menu(s)/Option(s)

The following options are exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288):

Option Name	Option Title	Description
XU-INSTITUTION-DEA	Institution DEA# edit	Kernel Patch XU*8.0*288— This option is used to edit the Facility DEA number in the INSTITUTION file (#4).
XUSSPKI CRL UPLOAD	PKI CRL Upload	Kernel Patch XU*8.0*283— This option calls CRLUP^XUSSPKI API.
XUSSPKI EDIT	Kernel PKI Parameter Edit	Kernel Patch XU*8.0*283— This option is used to enter or edit the Internet Protocol (IP) address of the Verification Server.

Table 3-8: Menu options exported with the ISS DEA/VA PKI Pilot Project

-  For more information on the options, please refer to the "Exported Options" section in this chapter.
-  For more information on the PKI Verification Server process, please refer to the "PKI Verification Server Process Diagram" topic in Chapter 1, "User Manual Information," in this manual.

Security Key(s)

There are *no* security keys exported with the ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288).

File Security

The following file security is established with ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288):

File #	File Name	DD	RD	WR	DEL	LAYGO	AUDIT
8980.2	PKI DIGITAL SIGNATURES	#	#	#	#	#	#
8980.22	PKI CRL URLS	#	#	#	#	#	#

Table 3-9: File security for the ISS DEA/VA PKI Pilot Project



Any attempts to edit/modify the data in these files will automatically invalidate the Digital Signature.

References

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) has been developed in accordance with DEA regulations.



Links to the DEA regulations are provided below:

Current DEA regulations online:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/index.html>

DEA regulations dealing with prescriptions:

<http://www.deadiversion.usdoj.gov/21cfr/cfr/2106cfrt.htm>

http://www.deadiversion.usdoj.gov/21cfr/cfr/1306/1306_05.htm

Official Policies

The ISS DEA/VA PKI Pilot Project (i.e., Kernel Patches XU*8.0*283 and 288) does *not* impose any additional legal requirements on VistA users, nor does it relieve users of any previously established requirements.

A Memorandum of Understanding (MOU) has been written and signed off of by both DEA and the VA.



The software and documentation provided with this project is only applicable through the ISS DEA/VA PKI Pilot Project "Development" and "Pilot" phases. Both the software and documentation are *not* ready for national release at this time. Upon completion of the DEA/VA PKI Pilot Project "Pilot" phase, both the software and documentation will be re-evaluated and updated as necessary, as well as the performance of any necessary tasks (e.g., Integration Agreements).

Glossary

CA	Certificate Authority
CAPI	Crypto Application Programming Interface (Microsoft Corporation)
CERTIFICATE REVOCATION LIST (CRL)	"A document maintained and published by a certification authority (CA) that lists certificates issued by the CA that are no longer valid." ²
CERTIFICATION AUTHORITY (CA)	<p>"A certification authority (CA) is an entity that creates and then 'signs' a document or file containing the name of a user and his public key. Anyone can verify that the file was signed by no one other than the CA by using the public key of the CA. By trusting the CA, one can develop trust in a user's public key.</p> <p>The trust in the certification authority's public key can be obtained recursively. One can have a certificate containing the certification authority's public key signed by a superior certification authority (<i>Root CA</i>) that he already trusts. Ultimately, one need only trust the public keys of a small number of top-level certification authorities. Through a chain of certificates (<i>Sub CAs</i>), trust in a large number of users' signatures can be established.</p> <p>A broader application of digital certification includes not only name and public key but also other information. Such a combination, together with a signature, forms an extended certificate. The other information may include, for example, electronic-mail address, authorization to sign documents of a given value, or authorization to sign other certificates."³</p>
CPRS	Computerized Patient Record System
CRL	Certificate Revocation List
CRYPTOAPI (CAPI)	Microsoft's "CryptoAPI (an application programming interface) provides services that enable application developers to add security based on cryptography to applications. CryptoAPI includes functionality for encoding to and decoding from ASN.1, hashing, encrypting and decrypting data, for authentication using digital certificates, and for managing certificates in certificate stores. Encryption and decryption are provided both using both session keys

² Microsoft's MSDN Library Web site (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/secglos_62nt.asp?frame=true): "Security, SDK Documentation (Platform SDK), Security Glossary."

³ DEA Web site (http://www.deadiversion.usdoj.gov/ecomme/erx/con_ops/index.html): "Public Key Infrastructure Analysis Concept of Operations," Section 3.4.3 "Public Key - The I in PKI"

Glossary

	and with public/private key pairs." ⁴
	CAPI is used to "cryptographically transform ordinary text (plaintext) into a coded form (ciphertext) by encryption and the transformation of ciphertext into plaintext by decryption in conjunction with PKI." ⁵
CS	Controlled Substance
CSP	Cryptographic Service Provider
DEA	Drug Enforcement Agency
DECRYPTION	"Cryptographic transformation of data (ciphertext) that restores encrypted data to its original state (plaintext)." ⁶
ENCRYPTION	"Cryptographic transformation of data (plaintext) into a form (ciphertext) that conceals the data's original meaning to prevent it from being known or used." ⁷
HIPAA	Health Insurance Portability and Accountability Act
NDF	National Drug File
PDM	Pharmacy Data Management
PEC	Performance Engineering Corporation Solutions Team
PKI	Public Key Infrastructure
PRIVATE CERTIFICATE	This is the certificate that contains both the user's public and private keys. This certificate will reside on a smart card.
PUBLIC CERTIFICATE	This is the certificate that contains the user's public key. This certificate will reside in a file or database.

4 Microsoft's MSDN Library Web site (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/security/portalsapi_3351.asp?frame=true): "Security, SDK Documentation (Platform SDK), Cryptography, CryptoAPI, Version 2.0, Purpose."

5 DEA Web site (http://www.dea.gov/ecomms/e_rx/con_ops/index.html): "Public Key Infrastructure Analysis Concept of Operations," Section 3.4.2 "Public Key - The PK in PKI"

6 DEA Web site (http://www.dea.gov/ecomms/e_rx/con_ops/index.html): "Public Key Infrastructure Analysis Concept of Operations," Section 3.4.1 "Terms and Definitions"

7 DEA Web site (http://www.dea.gov/ecomms/e_rx/con_ops/index.html): "Public Key Infrastructure Analysis Concept of Operations," Section 3.4.1 "Terms and Definitions"

PUBLIC KEY INFRASTRUCTURE (PKI)	<p>"PKI technology adds the following security services to an electronic ordering system:</p> <p>Confidentiality—only authorized persons have access to data.</p> <p>Authentication—establishes who is sending/receiving data.</p> <p>Integrity—the data has not been altered in transmission.</p> <p>Non-repudiation—parties to a transaction cannot convincingly deny having participated in the transaction."⁸</p>
ROOT CA	Root Certificate Authority
RX	Prescription
SMART CARD	<p>"An integrated circuit card (ICC) owned by an individual or a group whose information must be protected according to specific ownership assignments. It provides its own physical access control; without the smart card subsystem placing additional access control on the smart card.</p> <p>A smart card is a plastic card containing an integrated circuit that is compatible with ISO 7816."</p>
SUB CA	Subordinate Certificate Authority
THUMBPRINT	This is the foreign key (hash) used to access the public and private certificates.



For a comprehensive list of commonly used infrastructure- and security-related terms and definitions, please visit the ISS Glossary Web page at the following Web address:

<http://vista.med.va.gov/iss/glossary.asp>

For a list of commonly used acronyms, please visit the ISS Acronyms Web site at the following Web address:

<http://vista/med/va/gov/iss/acronyms/index.asp>

⁸ DEA Web site (http://www.deadiversion.usdoj.gov/ecom/e_rx/con_ops/index.html): "Public Key Infrastructure Analysis Concept of Operations," Section 3.3 "Security"

Glossary

Appendix A—API Error Management

The following table lists the possible API/COM object error codes that a user might encounter when signing or verifying a Schedule II prescription drug order. Also included are possible resolutions or suggested actions to correct the error:

Error Category	Error Code	Error Text	* Error Resolution/Suggested Action
Signing (CPRS)	89802000	Order text to be signed is empty.	The data buffer for signing is empty. This should never happen; the CPRS software should catch this error before calling the ISS Signing COM object. IRM should enter a NOIS for CPRS.
Signing (CPRS)	89802001	User's DEA # is missing.	The user's DEA number needs to be added to the NEW PERSON file (#200). This should never happen; the CPRS software should catch this error before calling the ISS Signing COM object. IRM should enter a NOIS for CPRS.
Signing (CPRS)	89802002	Drug Schedule is missing.	The DEA Drug Schedule was not set before calling the signing procedure. This should never happen; the CPRS software should catch this error before calling the ISS Signing COM object. IRM should enter a NOIS for CPRS.
Signing (CPRS)	89802003	No cert with a valid date was found.	The ISS Signing COM object did not find any certificate that was valid for the current date (see workaround below). IRM should check to see if the DEA certificate is registered on the client workstation being used to digitally sign the prescription. Use the Microsoft Windows utility to view certificates (see the "Viewing Certificates on a Client Workstation" bullet under the "Certification Application Process" topic in Chapter 1, "User Manual Information," in this manual. If the certificate is not registered, have the user re-insert the smart card into the reader to register the certificate.
Signing (CPRS)	89802004	Valid Certificate was not found.	The ISS Signing COM object could not find a valid certificate. Thus, the prescription cannot be digitally signed (see workaround below).

Error Category	Error Code	Error Text	* Error Resolution/Suggested Action
Signing (CPRS)	89802005	Couldn't load CSP.	<p>There was a problem trying to load the Cryptographic Service Provider (CSP) COTS software.</p> <p>IRM should verify that the CSP Passage software and smart card reader were properly installed on the client workstation being used to digitally sign a prescription.</p>
Signing (CPRS)	89802006	Smart Card Reader not found.	IRM should verify that the smart card reader was properly installed on the client workstation being used to digitally sign a prescription.
Signing (CPRS)	89802007	Cert with DEA # not found.	<p>The ISS Signing COM object could not find a certificate with the user's DEA number. Thus, the prescription cannot be digitally signed (see workaround below).</p> <p>IRM should check to see if the user's DEA certificate is registered on the client workstation being used to digitally sign the prescription. Use the Microsoft Windows utility to view certificates (see the "Viewing Certificates on a Client Workstation" bullet under the "Certification Application Process" topic in Chapter 1, "User Manual Information," in this manual. If the certificate is not registered, have the user re-insert the smart card into the reader to register the certificate.</p>
Signing (CPRS)	89802008	Cert not valid for Drug schedule.	The holder of this certificate is <i>not</i> authorized to prescribe the Schedule of this drug.
Signing (CPRS)	89802009	Signature Check failed (Invalid Signature).	A test of the just created signature failed.

Error Category	Error Code	Error Text	* Error Resolution/Suggested Action
Signing (CPRS) and Verification (Pharmacy)	89802010	Crypto error (contact IRM).	<p>This is a general error message.</p> <p>IRM should first check the memory statistics and reboot the client workstation being used to digitally sign a prescription.</p> <p>If the error occurs during the signature verification process, IRM should check for the following conditions:</p> <ul style="list-style-type: none"> • See that the PKIserver.exe routine is installed on the server in the ..\WINNT\system32 directory. • See that the PKIserver.exe routine has been registered with Windows. This is done via: <p style="margin-left: 40px;">Start>Run C:\WINNT\system32\ PKIserver /Install</p> • See that in Administrator tools>services that PKI_Verify_Service has a status of STARTED and its Startup Type is Automatic.
Signing (CPRS)	89802011	Certificate Chain not valid.	One of the certificates in the trust chain is not valid. It may not have been installed on this computer, or it may have expired.
Verification (Pharmacy)	89802015	Signature failed: Corrupted (Decode failure).	<p>The decoding of the Digital Signature has failed, because the Digital Signature is corrupted. Thus, the digital signature for the Schedule II prescription cannot be verified.</p> <p>This should only happen if the data has been modified or tampered with so that it is no longer valid. The user should re-try the transaction. If the transaction continues to fail, IRM should enter a NOIS for the ISS Verification API (and COTS) software.</p>

Error Category	Error Code	Error Text	* Error Resolution/Suggested Action
Verification (Pharmacy)	89802016	Signature failed: Corrupted (Hash mismatch).	<p>The two hash values do not match. Thus, the digital signature for the Schedule II prescription cannot be verified.</p> <p>This should only happen if the data has been modified or tampered with so that it is no longer valid. The user should re-try the transaction. If the transaction continues to fail, IRM should enter a NOIS for the ISS Verification API (and COTS) software.</p>
Verification (Pharmacy)	89802017	Signature failed: Certificate revoked.	<p>The certificate is on a Certificate Revocation List (CRL). Thus, the certificate used to digitally sign the prescription has been revoked.</p> <p>Pharmacy should evaluate.</p>
Verification (Pharmacy)	89802018	Signature failed: Verification failure.	<p>The Digital Signature for the Schedule II prescription cannot be verified.</p> <p>Pharmacy should evaluate.</p>
Verification (Pharmacy)	89802019	Signature failed: Before Cert effective date.	<p>The digital signature of the prescription is dated before the effective date of the certificate.</p> <p>This should never happen. IRM should check the system time on both the client workstation that was used to digitally sign the prescription and the VistA M Server being used to verify the digital signature.</p>
Verification (Pharmacy)	89802020	Signature failed: Certificate expired.	<p>The certificate used to digitally sign the Schedule II prescription is expired.</p> <p>Pharmacy should evaluate.</p>

Table A-1: PKI API error codes and their resolutions

***Workaround**—For all PKI API-related errors, sites should follow the normal established procedures for signing and verifying Schedule II prescriptions prior to the implementation of the PKI Digital Signature software.



For more detailed descriptions of these error messages, please refer to the DESCRIPTION field in the DIALOG file (#.84).

Index

A

Acknowledgements, xi
ACTIVE by Custodial Package Option, 3-15
Administrative Tasks, 1-5
 Introduction, 1-5
 Overview Diagram, 1-9
APIs
 \$\$DEA^XUSER (DEA Number API), 2-8
 \$\$STORESIG^XUSSPKI (PKI Data Storage API), 2-6
 \$\$VERIFY^XUSSPKI (Digital Signature Verification API), 2-7
 Error Management, Appendix A, 1
 IXuDigSigS (Digital Signing COM), 2-1
Appendix A—API Error Management, 1
Application Entry Points, 3-12
Application Program Interfaces (APIs), 2-1
 Error Management, Appendix A, 1
Architecture Broad Overview Diagram, 1-4
Archiving and Purging, 3-11, 3-20
Assumptions About the Reader, xv

B

Bulletins, 3-3

C

Callable Routines, 3-12
Certificate Revocation List (CRL), 1-5, 1-16
Certificates
 Application Process, 1-6
 Certificate Revocation List (CRL), 1-5, 1-16
 Data, 1-7
 Locating DEA Certificate, 1-11
 Maintaining, 1-8
 Populating Smart Card, 1-7
 Retrieving Stored Certificate, 1-16
 Sub CA Certificate Application, 1-6
 VA Certificate Application, 1-6
 Validating DEA Certificate, 1-11
 Viewing on a Client Workstation, 1-7
Certification
 Application Process, 1-6
 Maintenance Process, 1-8
Check
 For Smart Card, 1-11
 The Trust Chain, 1-16

COM

IXuDigSigS (Digital Signing COM), 2-1
Communications Interfaces, 3-13
Connect to PKI Verification Server, 1-16
Contents, v
Controlled Subscription
 Integration Agreements, 2-1
 References, 2-1, 3-14
Custodial Package Menu, 3-15

D

Data Dictionary
 Data Dictionary Utilities Menu, xv
 Listings, xv
DBA IA CUSTODIAL MENU, 3-15
DBA IA CUSTODIAL Option, 3-15
DBA IA INQUIRY Option, 3-15
DBA IA ISC Menu, 3-15, 3-16
DBA IA SUBSCRIBER MENU, 3-16
DBA IA SUBSCRIBER Option, 3-16
DBA Menu, 3-15, 3-16
DEA
 Authorization Process, 1-6
 Regulations, 1-12
 Section 1306.05 Manner of issuance of prescriptions, 1-13
 Web Links, 1-13, 3-11, 3-20, 3-23
DEA Number API
 \$\$DEA^XUSER, 2-8
DEA# Field (#53.2), 2-8
Dependencies, 3-14
Diagrams
 Administrative Tasks Overview, 1-9
 Architecture Broad Overview, 1-4
 Signing Functionality Overview, 1-14
 Verification
 Functionality Overview, 1-18
 Server Process, 1-19
Digial Signature and Prescription Data, 1-12
Digital Signature Verification API
 \$\$VERIFY^XUSSPKI, 2-7
Digital Signature(s), 3-21
Digital Signing COM
 IXuDigSigS, 2-1
Digital Signing COM Process, 1-11
Digitally Sign a Block of Data, 1-11
Directives, 3-23

Disconnect from PKI Verification Server, 1-16

Documentation

History, iii

Symbols, xiii

E

Enter Prescription Data, 1-10

Entry Points, 3-12

Error Management

Application Program Interfaces (APIs),

Appendix A, 1

Exported Options, 3-7

External

Interfaces, 3-13

Relations, 3-14

F

FACILITY DEA NUMBER Field (#52), 1-2, 3-6

Fields, 3-6

DEA# (#53.2), 2-8

FACILITY DEA NUMBER (#52), 1-2, 3-6

PKI SERVER ADDRESS (#53.1), 3-6

Figures and Tables, ix

Files

Fields, 3-6

Global, 3-5

INSTITUTION (#4), 1-2, 3-6, 3-7, 3-8, 3-22

KERNEL SYSTEM PARAMETERS

(#8989.3), 3-6, 3-9

List, 3-6

NEW PERSON (#200), 1-2, 2-8

Numbers, Global Locations, 3-18

PKI CRL URLS (#8980.22), 3-6

Security, 3-22

PKI DIGITAL SIGNATURES (#8980.2), 2-6,

3-6, 3-11, 3-20

Security, 3-22

Security, 3-22

G

G&D Smart Cards, 1-6, 1-7, 3-13, 3-21

General Instructions for Obtaining IAs on

FORUM, 3-15

Global, 3-5

^XUSSPKI, 3-5

Locations, 3-18

Global and Files List, 3-5

Glossary, 1

H

Hardware Interfaces, 3-13

Help

At Prompts, xiv

Online, xiv

History, Revisions to Documentation and

Patches, iii

Home Pages

Adobe Acrobat Quick Guide Web Address, xvi

Adobe Web Address, xvi

CPRS Documentation Web Address, 1-10, 1-12

DEA Regulations Web Address, 1-13, 3-23

Health Systems Design and Development (HSD&D) Web Address, xv

ISS PKI Pilot Project Home Page Web Address, xv

Kernel Home Page Web Address, xvi

Pharmacy Documentation Web Address, 1-15, 1-17

RSA Web Address, 3-13, 3-21

VA PKI Pilot Projects Home Page Web Address, xvi

VA/DEA PKI Pilot Project Home Page Web Address, xvi

How to

Generate Technical Information Online, xiv

Use this Manual, xiii

I

Implementation, 3-1

Inquire Option, 3-15

Installation Instructions

Kernel Patches XU*8.0*283 and 288, 3-1

Institution DEA# edit Option, 3-7, 3-8, 3-22

INSTITUTION File (#4), 1-2, 3-6, 3-7, 3-8, 3-22

Instructions

Installation of Kernel Patches XU*8.0*283 and 288, 3-1

Post Installation, 3-1

Integration Agreements (IAs), 3-14

Controlled Subscription References, 2-1, 3-14

General Instructions for Obtaining IAs from FORUM, 3-15

Integration Agreements Menu, 3-15, 3-16

Supported References, 2-8

Interfaces

Communication, 3-13

Hardware, 3-13
 Software, 3-13
 Interfacing, 3-21
 Internal Relations, 3-17
 Introduction
 Administrative Tasks, 1-5
 PKI Signing Functionality, 1-10
 User Manual Information, 1-1
 Verification Functionality, 1-15
 Invoke COTS Software, 1-16
 ISS DEA/VA PKI Pilot Project
 APIs
 \$\$DEA^XUSER (DEA Number API), 2-8
 \$\$STORESIG^XUSSPKI (PKI Data
 Storage API), 2-6
 \$\$VERIFY^XUSSPKI (Digital Signature
 Verification API), 2-7
 Error Management, Appendix A, 1
 IXuDigSigS (Digital Signing COM), 2-1
 Application Program Interfaces (APIs), 2-1
 Error Management, Appendix A, 1
 Namespace, 3-17
 Purpose, 1-2
 Scope, 1-3

K

Kernel
 APIs for the ISS DEA/VA PKI Pilot Project
 \$\$DEA^XUSER (DEA Number API), 2-8
 \$\$STORESIG^XUSSPKI (PKI Data
 Storage API), 2-6
 \$\$VERIFY^XUSSPKI (Digital Signature
 Verification API), 2-7
 Error Management, Appendix A, 1
 IXuDigSigS (Digital Signing COM), 2-1
 Home Page Web Address, xv
 Namespace, 3-17
 Kernel Management Menu, 3-7
 Kernel PKI Parameter Edit Option, 3-2, 3-7, 3-9,
 3-22
 KERNEL SYSTEM PARAMETERS File
 (#8989.3), 3-6, 3-9

L

List File Attributes Option, xv
 Locate DEA Certificate, 1-11

M

Mail Group Edit Option, 3-2, 3-3

Mail Groups, 3-3, 3-20
 XUSSPKI CRL SERVER, 3-2, 3-3
 Maintenance, 3-3
 Manual Organization, xiii
 Memory Constraints, 3-2
 Menu(s)/Option(s)
 Security, 3-22
 Menus
 Custodial Package Menu, 3-15
 Data Dictionary Utilities, xv
 DBA, 3-15, 3-16
 DBA IA CUSTODIAL MENU, 3-15
 DBA IA ISC, 3-15, 3-16
 DBA IA SUBSCRIBER MENU, 3-16
 Integration Agreements Menu, 3-15, 3-16
 Kernel Management, 3-7
 Operations Management, 3-7
 Security, 3-22
 Subscriber Package Menu, 3-16
 XUKERNEL, 3-7
 XUSITEMGR, 3-7

N

Namespace, 3-17
 NEW PERSON File (#200), 1-2, 2-8

O

Official Policies, 3-23
 Online
 Documentation, xiv
 Help Frames, xiv
 Technical Information, How to Generate, xiv
 Operations, 3-2
 Operations Management Menu, 3-7
 Option(s)/Menu(s)
 Security, 3-22
 Options
 ACTIVE by Custodial Package, 3-15
 DBA IA CUSTODIAL, 3-15
 DBA IA INQUIRY, 3-15
 DBA IA SUBSCRIBER, 3-16
 Exported, 3-7
 Inquire, 3-15
 Institution DEA# edit, 3-7, 3-8, 3-22
 Kernel Management, 3-7
 Kernel PKI Parameter Edit, 3-2, 3-7, 3-9, 3-22
 List File Attributes, xv
 Mail Group Edit, 3-2, 3-3
 Operations Management, 3-7
 PKI CRL Upload, 3-2, 3-10, 3-22

Print ACTIVE by Subscribing Package, 3-16
 Schedule/Unschedule Options, 3-2
 Security, 3-22
With Parents, 3-7
Without Parents, 3-10
 XMEDITMG, 3-2, 3-3
 XU-INSTITUTION-DEA, 3-7, 3-8, 3-22
 XUKERNEL, 3-7
 XUSITEMGR, 3-7
 XUSSPKI CRL UPLOAD, 3-2, 3-10, 3-22
 XUSSPKI EDIT, 3-2, 3-7, 3-9, 3-22

Organization of the Manual, xiii

Orientation, xiii

P

Passage Software

RSA, 1-6, 1-7, 1-11, 2-2, 2-4, 3-13, 3-21

Patches

History, iii

PKI

ISS PKI Pilot Project Home Page Web

Address, xv

VA PKI Pilot Projects

Home Page Web Address, xvi

VA/DEA PKI Pilot Project

Home Page Web Address, xvi

PKI CRL Upload Option, 3-2, 3-10, 3-22

PKI CRL URLS File (#8980.22), 3-6

Security, 3-22

PKI Data Storage API

\$\$STORESIG^XUSSPKI, 2-6

Process, 1-12

PKI DIGITAL SIGNATURES File (#8980.2),

2-6, 3-6, 3-11, 3-20

Security, 3-22

PKI SERVER ADDRESS Field (#53.1), 3-6, 3-

9

PKI Software

Signing Functionality, 1-10

Verification Functionality, 1-15

PKI Verification

Server Process Diagram, 1-19

Policies, Official, 3-23

Post Installation Instructions, 3-1

Prepare for Prescription Entry, 1-10

Prescription and Digital Signature Data, 1-12

Print ACTIVE by Subscribing Package Option,
 3-16

Processes

Certification Application, 1-6

Certification Maintenance, 1-8

DEA Authorization, 1-6

Digital Signing COM, 1-11

PKI Data Storage API, 1-12

Signing Functionality, 1-10

VA Pharmacist Process, 1-15

VA Practitioner, 1-10

Verification, 1-15

Verifying Digital Signature API, 1-16

Viewing Certificates on a Client Workstation,
 1-7

VistA CPRS Software (Part I), 1-10

VistA CPRS Software (Part II), 1-12

VistA Pharmacy Software, 1-15

Programmer Manual Information, 2-1

Purging and Archiving, 3-11, 3-20

Purpose of the ISS DEA/VA PKI Pilot Project,
 1-2

Q

Question Mark Help, xiv

R

Reader, Assumptions About the, xv

Reference Materials, xv

References, 3-23

Controlled Subscription, 2-1, 3-14

General Instructions for Obtaining IAs from
 FORUM, 3-15

Supported, 2-8

Remote System(s), 3-20

Required Administrative Components and
 Processes, 1-5

Retrieve

Patient Record, 1-10

Stored Certificate, 1-16

Return

Digital Signature and Hash to VistA Calling
 Application, 1-11

Response to

VistA API, 1-16

VistA Calling Application, 1-16

Revision History, iii

Documentation, iii

Patches, iii

Root Certification Authority, 1-5

Routines, 3-4

XUSC1, 3-4

XUSC1C, 3-4

XUSC1S, 3-4

XUSC1S1, 3-4
 XUSSPKI, 3-4

S

Schedule/Unschedule Options Option, 3-2
 Scope of the ISS DEA/VA PKI Pilot Project, 1-3
 Section 1306.05 Manner of issuance of prescriptions, 1-13
 Security, 3-20
 Archiving and Purging, 3-20
 Digital Signature(s), 3-21
 Files, 3-22
 Interfacing, 3-21
 Key(s), 3-22
 Mail Groups, 3-20
 Menu(s)/Option(s), 3-22
 Official Policies, 3-23
 References, 3-23
 Remote System(s), 3-20
 Signing
 Functionality, 1-10
 Introduction, 1-10
 Overview Diagram, 1-14
 Step-By-Step Procedures, 1-10
 Smart Card
 Certificate Population, 1-7
 Data Population, 1-6
 Distribution, 1-7
 G&D Smart Cards, 1-6, 1-7, 3-13, 3-21
 Readers, 1-6, 1-7, 3-13, 3-21
 Software
 Product Security, 3-20
 Requirements, 3-14
 Software Interfaces, 3-13
 Software-wide Variables, 3-19
 Special Operations, 3-2
 Step-By-Step Procedures
 Signing, 1-10
 Verification, 1-15
 Viewing Certificates on a Client Workstation, 1-7
 Store Digital Signature and Hash, 1-12
 Sub CA, 1-16
 Certificate Application, 1-6
 Subordinate Certification Authority, 1-5
 Subscriber Package Menu, 3-16
 Supported
 References, 2-8
 Symbols Found in the Documentation, xiii

T

Technical Manual Information, 3-1

U

URLs
 Adobe Acrobat Quick Guide Web Address, xvi
 Adobe Home Page Web Address, xvi
 CPRS Documentation Web Address, 1-10, 1-12
 DEA Regulations Web Address, 1-13, 3-23
 Health Systems Design and Development (HSD&D) Home Page Web Address, xv
 ISS PKI Pilot Project Home Page Web Address, xv
 Kernel Home Page Web Address, xvi
 Pharmacy Documentation Web Address, 1-15, 1-17
 RSA Home Page Web Address, 3-13, 3-21
 VA PKI Pilot Projects Home Page Web Address, xvi
 VA/DEA PKI Pilot Project Home Page Web Address, xvi
 Use this Manual, How to, xiii
 User Manual Information, 1-1
 Introduction, 1-1

V

VA Certificate Application, 1-6
 VA Pharmacist Process, 1-15
 VA Practitioner Process, 1-10
 Validate DEA Certificate, 1-11
 Variables, 3-19
 Verification
 Functionality, 1-15
 Introduction, 1-15
 Overview Diagram, 1-18
 Server Process Diagram, 1-19
 Step-By-Step Procedures, 1-15
 Verifying Digital Signature API Process, 1-16
 Viewing Certificates on a Client Workstation, 1-7
 VistA CPRS Software Process
 Part I, 1-10
 Part II, 1-12
 VistA Pharmacy Software Process, 1-15

W

Web Pages

Index

Adobe Acrobat Quick Guide Web Address,
xvi
Adobe Home Page Web Address, xvi
CPRS Documentation Web Address, 1-10, 1-
12
DEA Regulations Web Address, 1-13, 3-23
Health Systems Design and Development
(HSD&D) Home Page Web Address, xv
ISS PKI Pilot Project Home Page Web
Address, xv
Kernel Home Page Web Address, xvi
Pharmacy Documentation Web Address, 1-15,
1-17
RSA Home Page Web Address, 3-13, 3-21
VA PKI Pilot Projects Home Page Web
Address, xvi
VA/DEA PKI Pilot Project Home Page Web
Address, xvi

X

XMEDITMG Option, 3-2, 3-3
XU-INSTITUTION-DEA Option, 3-7, 3-8, 3-22
XUKERNEL Menu, 3-7
XUSC1 Routine, 3-4
XUSC1C Routines, 3-4
XUSC1S Routine, 3-4
XUSC1S1 Routine, 3-4
XUSITEMGR Menu, 3-7
XUSSPKI CRL SERVER Mail Group, 3-2, 3-3,
3-20
XUSSPKI CRL UPLOAD Option, 3-2, 3-10, 3-
22
XUSSPKI EDIT Option, 3-2, 3-7, 3-9, 3-22
XUSSPKI Global, 3-5
XUSSPKI Routine, 3-4